nature portfolio

https://doi.org/10.1038/s41596-023-00810-1

Supplementary information

Preparation and quantitative analysis of multicenter luminescence materials for sensing function

In the format provided by the authors and unedited

Supplementary Information for

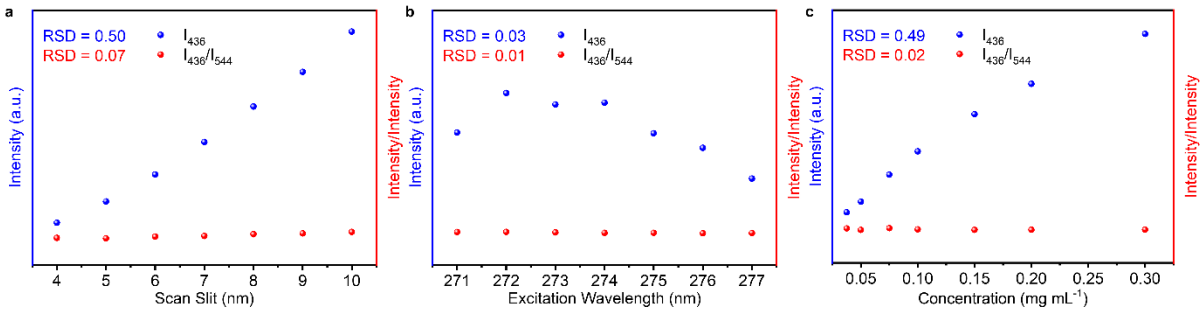
Preparation and Quantitative Analysis of Multi-center Luminescence Materials for Sensing Function

Zongsu Han[1], Kunyu Wang[2], Hong-Cai Zhou[2], Peng Cheng[1,3] and Wei Shi[1,3*]

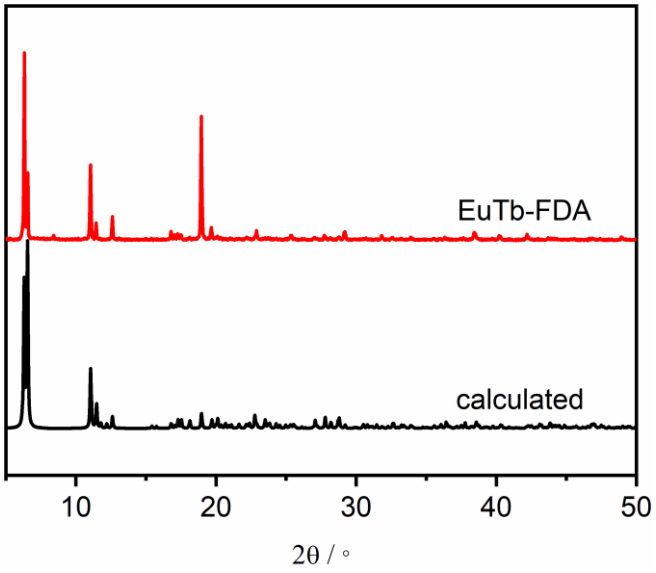
[1] Department of Chemistry, Key Laboratory of Advanced Energy Materials Chemistry (MOE) and Renewable Energy Conversion and Storage Center (RECAST), College of Chemistry, Nankai University, Tianjin, China. *e-mail: shiwei@nankai.edu.cn

[2] Department of Chemistry, Texas A&M University, College Station, Texas, United States.

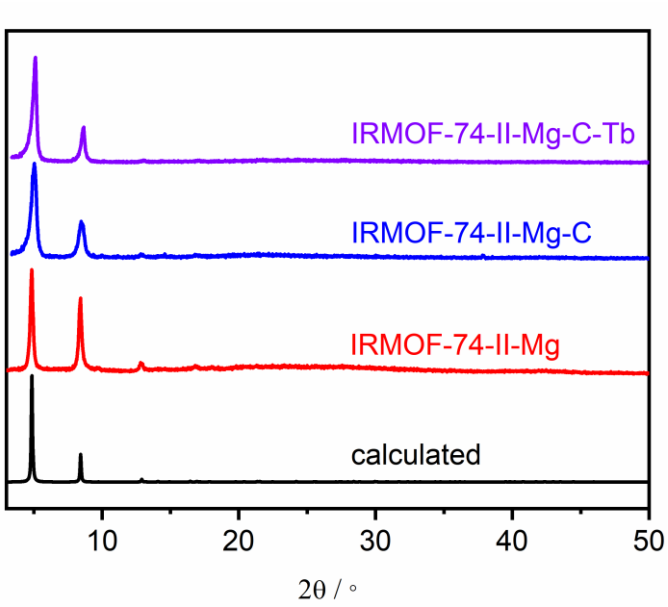
[3] Haihe Laboratory of Sustainable Chemical Transformations, Tianjin, China.



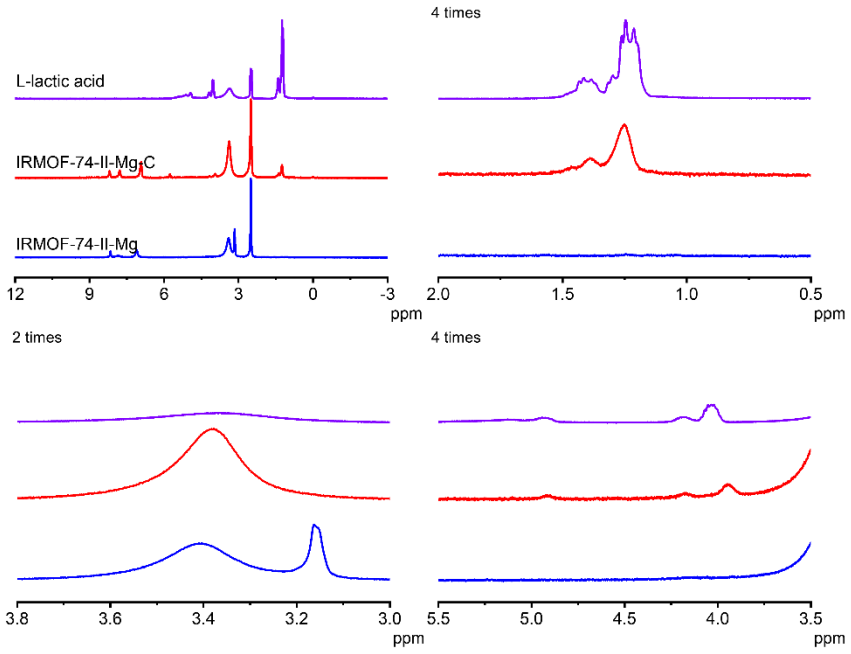
Supplementary Figure 1 Control experiments for IRMOF-74-II-Mg-C-Tb. Reprinted with permission from ref. 1, Wiley-VCH.



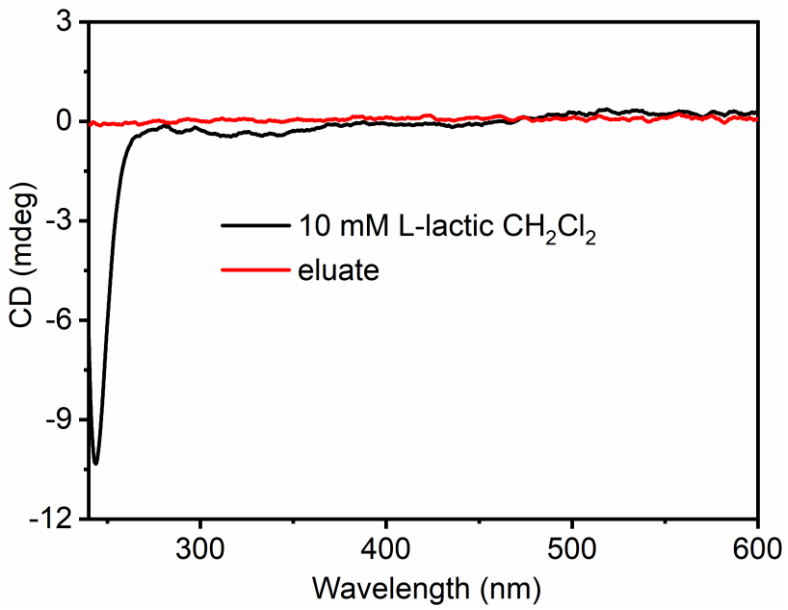
Supplementary Figure 2 PXRD for EuTb-FDA.



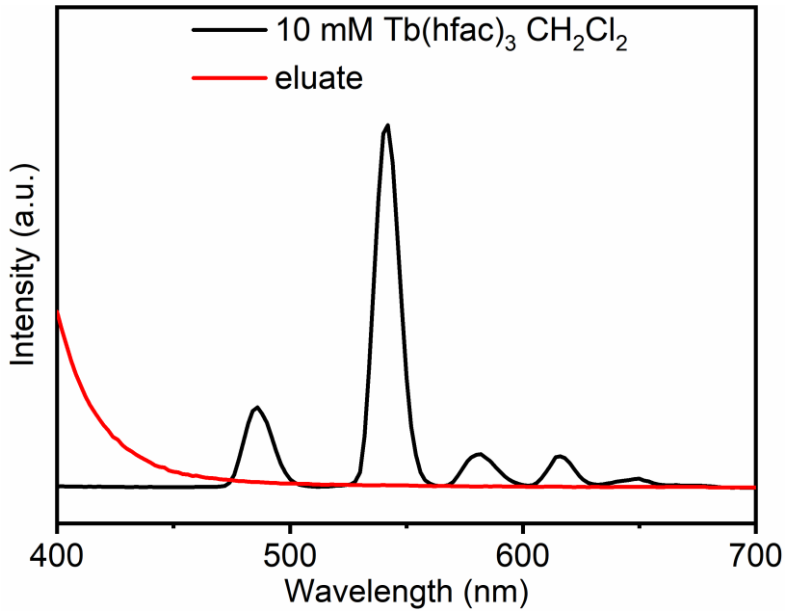
Supplementary Figure 3 PXRD for IRMOF-74-II-Mg, IRMOF-74-II-Mg-C and IRMOF-74-II-Mg-C-Tb. Reprinted with permission from ref. 1, Wiley-VCH.



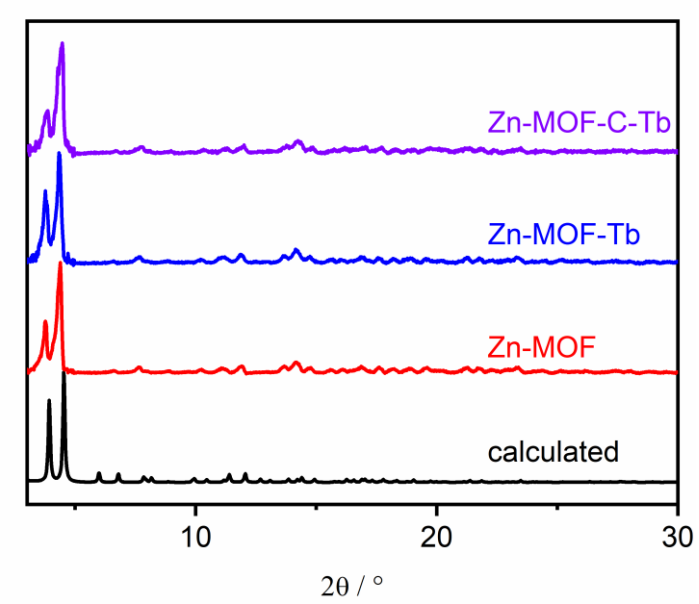
Supplementary Figure 4 ^{1}H NMR spectra of IRMOF-74-II-Mg and IRMOF-74-II-Mg-C. Reprinted with permission from ref. 1, Wiley-VCH.



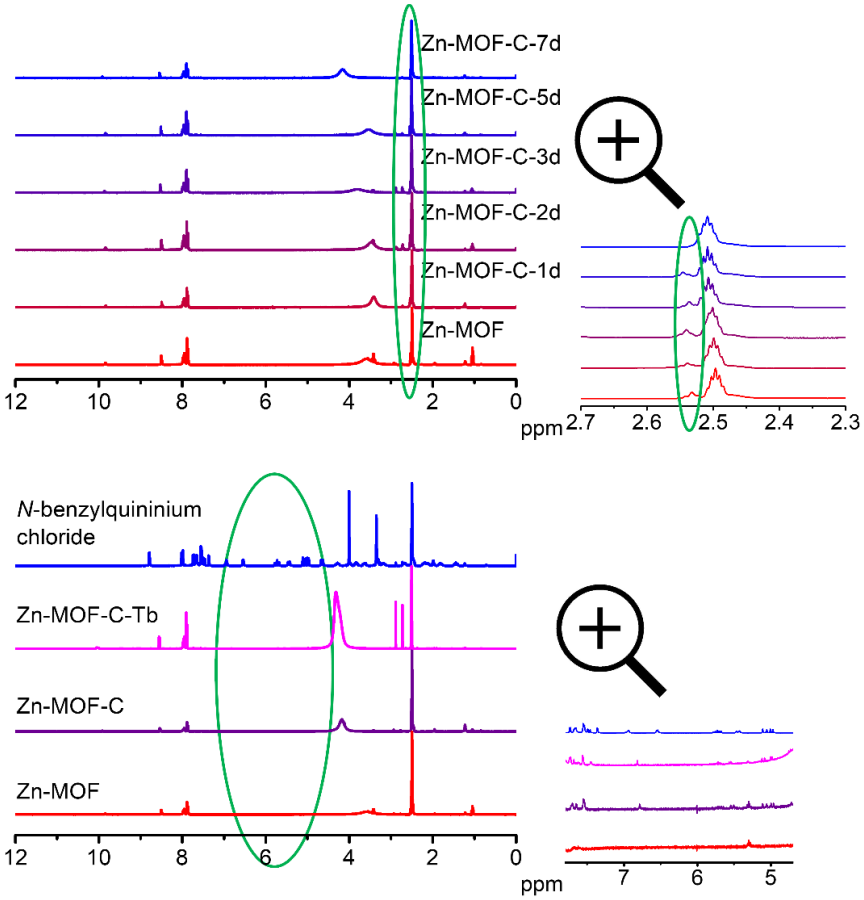
Supplementary Figure 5 CD spectra of the eluate of IRMOF-74-II-Mg-C. Reprinted with permission from ref. 1, Wiley-VCH.



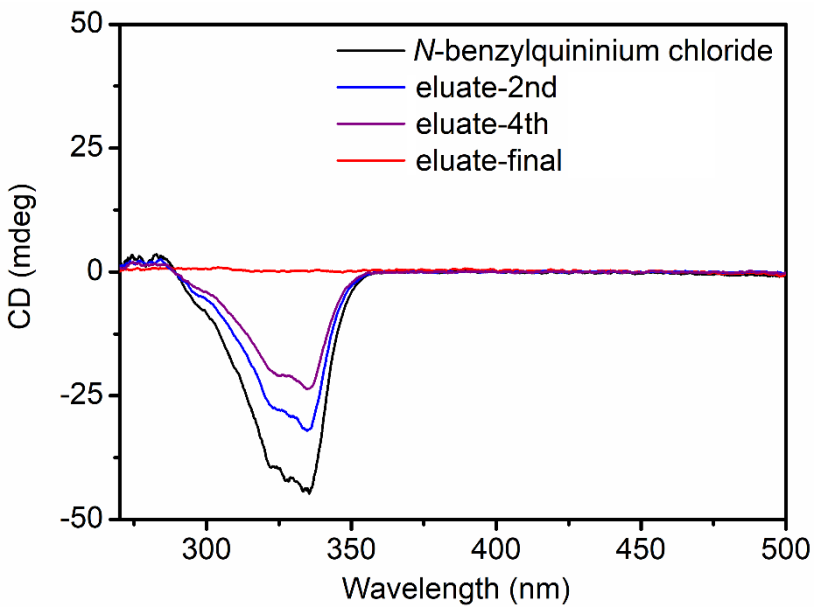
Supplementary Figure 6 Emission spectrum of the eluate of IRMOF-74-II-Mg-C. Reprinted with permission from ref. 1, Wiley-VCH.



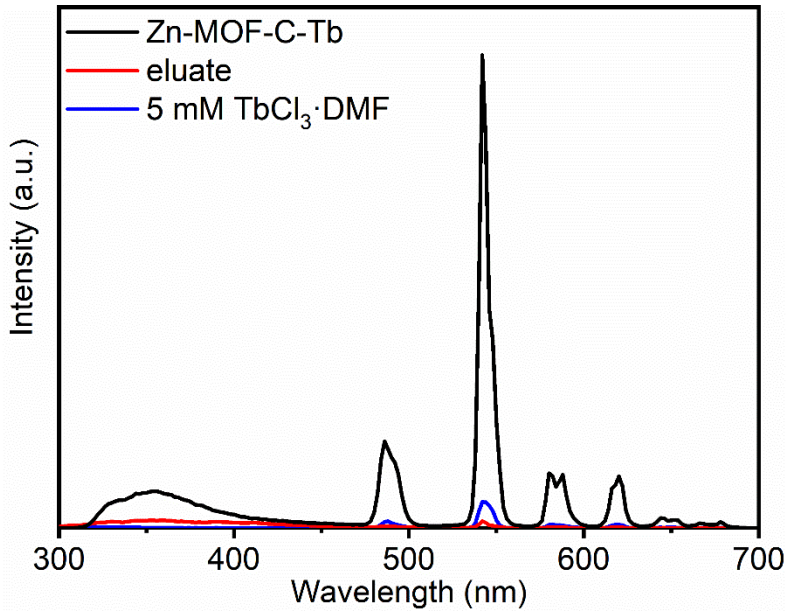
Supplementary Figure 7 PXRD for Zn-MOF, Zn-MOF-C and Zn-MOF-C-Tb. Reprinted with permission from ref. 2, Nature.



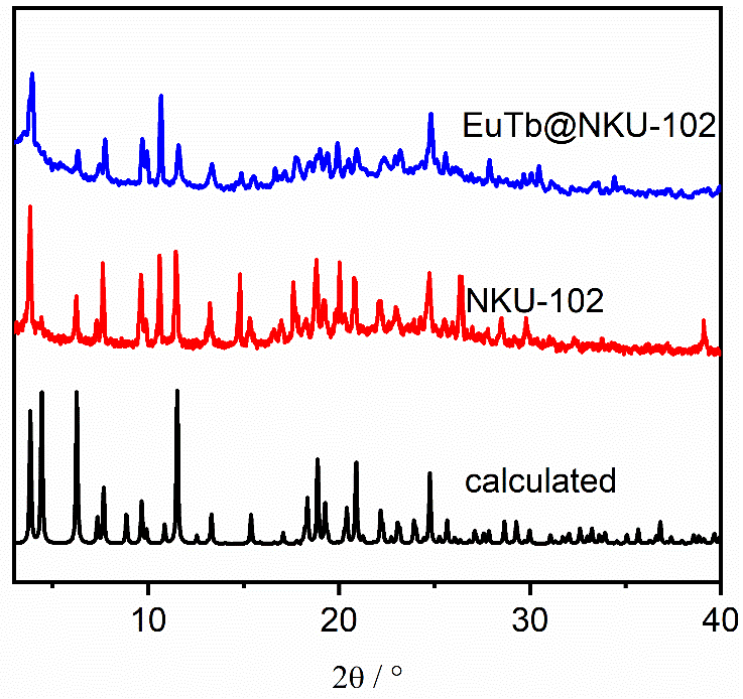
Supplementary Figure 8 ^{1}H NMR spectra of Zn-MOF and Zn-MOF-C. Reprinted with permission from ref. 2, Nature.



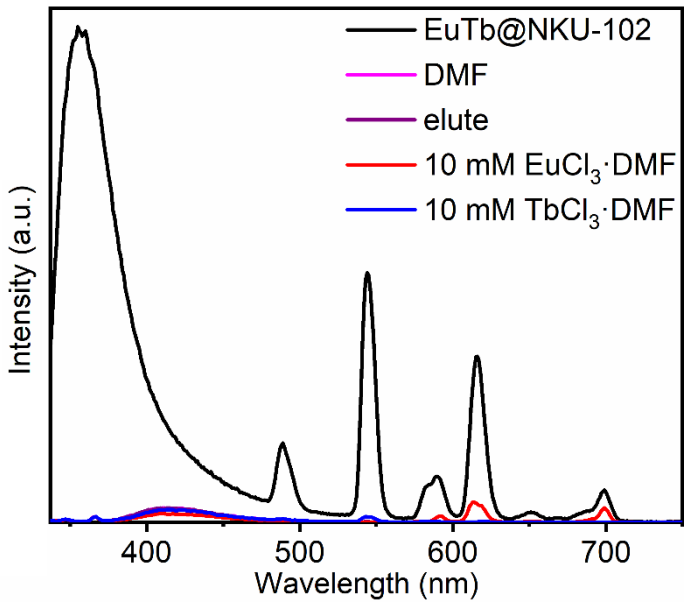
Supplementary Figure 9 CD spectra of the eluate of Zn-MOF-C. Reprinted with permission from ref. 2, Nature.



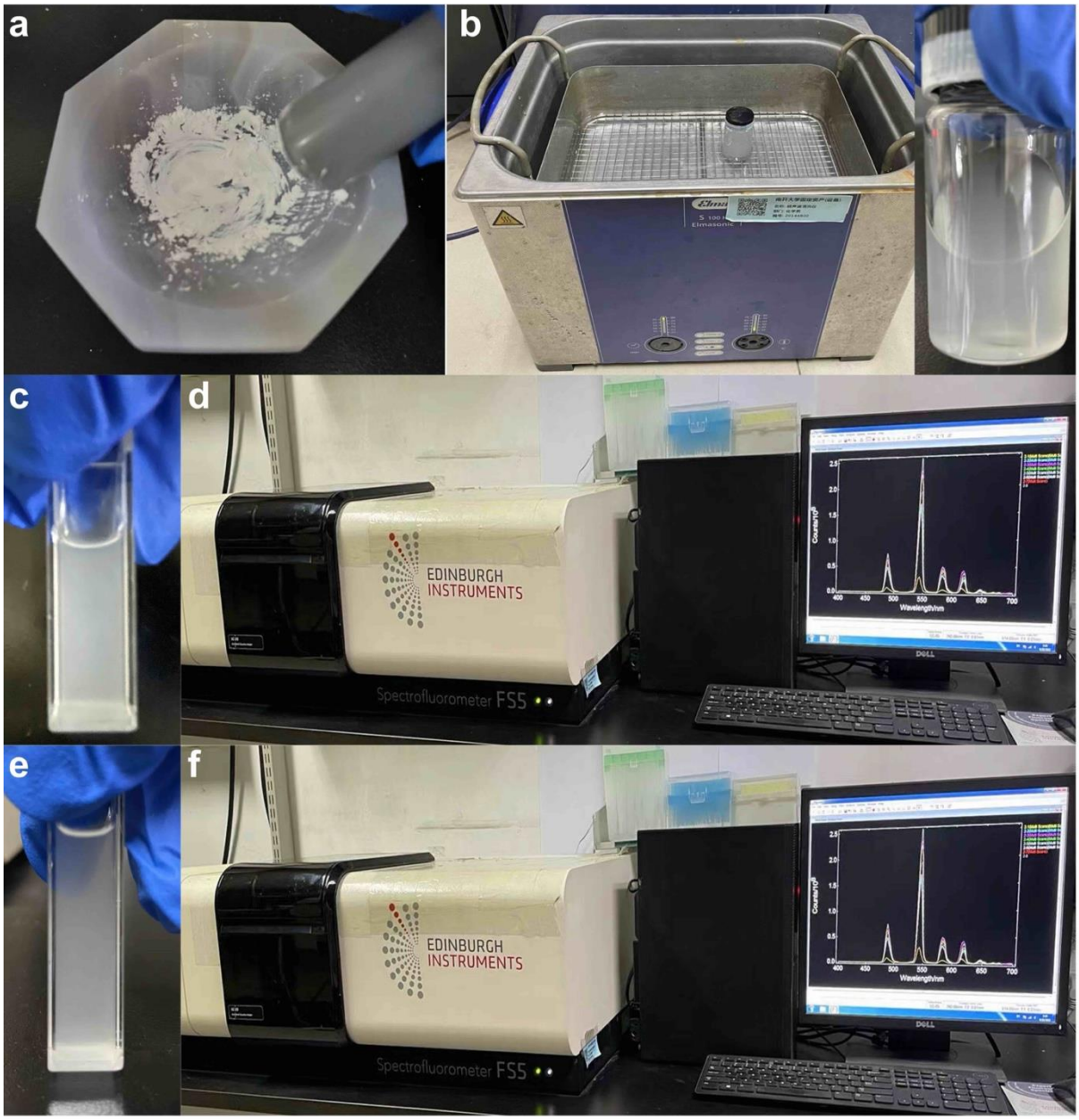
Supplementary Figure 10 Emission spectrum of the eluate of Zn-MOF-C-Tb. Reprinted with permission from ref. 2, Nature.



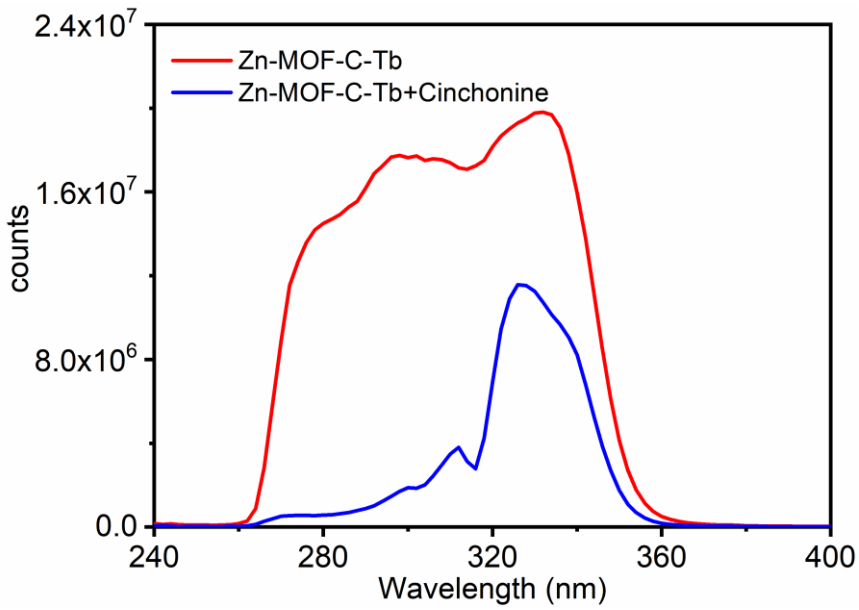
Supplementary Figure 11 PXRD for NKU-102 and EuTb@NKU-102. Reprinted with permission from ref. 3, Chinese Chemical Society.



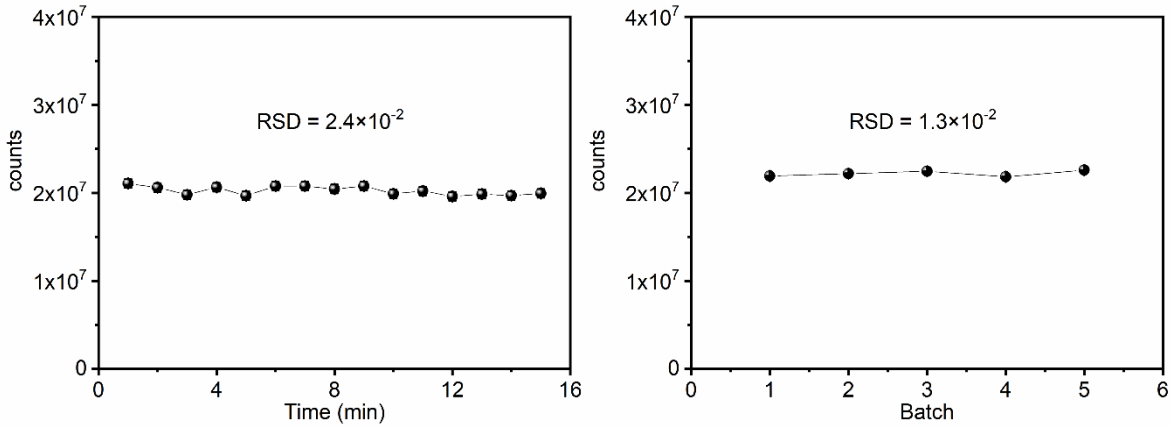
Supplementary Figure 12 Emission spectra of the eluate of EuTb@NKU-102. Reprinted with permission from ref. 3, Chinese Chemical Society.



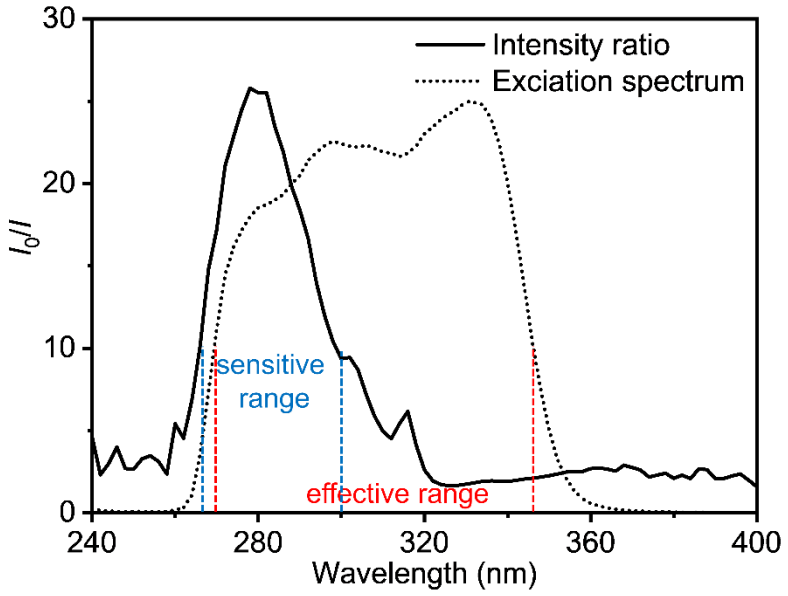
Supplementary Figure 13 Main detection procedure with instruments. a. Grind the samples into fine powder. b. Disperse samples under ultrasound. c. Move the dispersions into the fluorescence cuvette. d. Collect the emission spectrum of the samples. e. Add the analytes. f. Collect the emission spectrum of the samples with analytes.



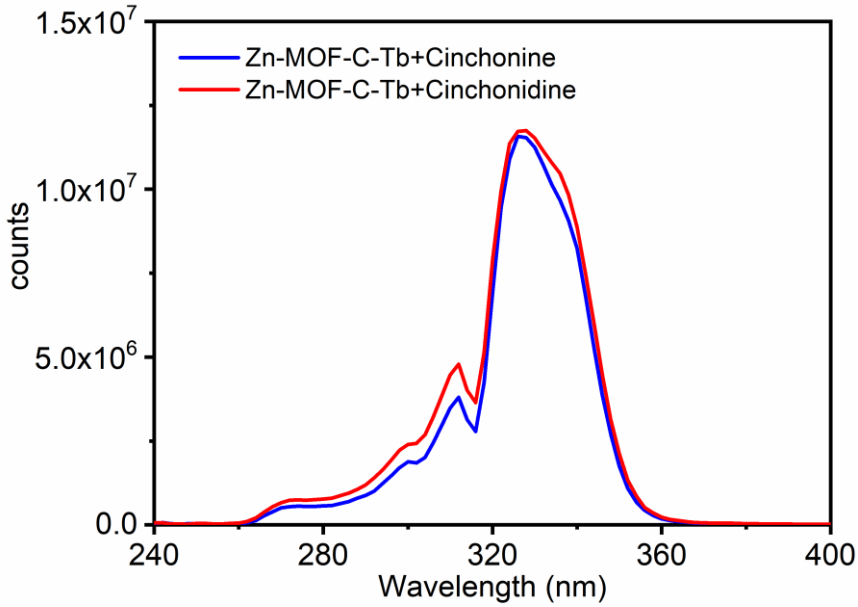
Supplementary Figure 14 Excitation spectra of Zn-MOF-C-Tb and Zn-MOF-C-Tb with Cinchonine at 544 nm. Reprinted with permission from ref. 2, Nature.



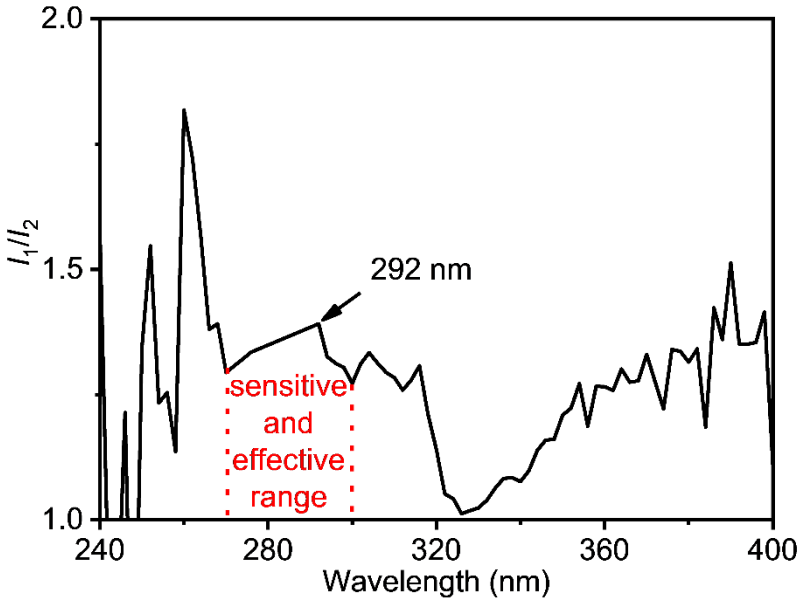
Supplementary Figure 15 Time-dependent and different-batches of luminescent intensities at 544 nm. Reprinted with permission from ref. 2, Nature.



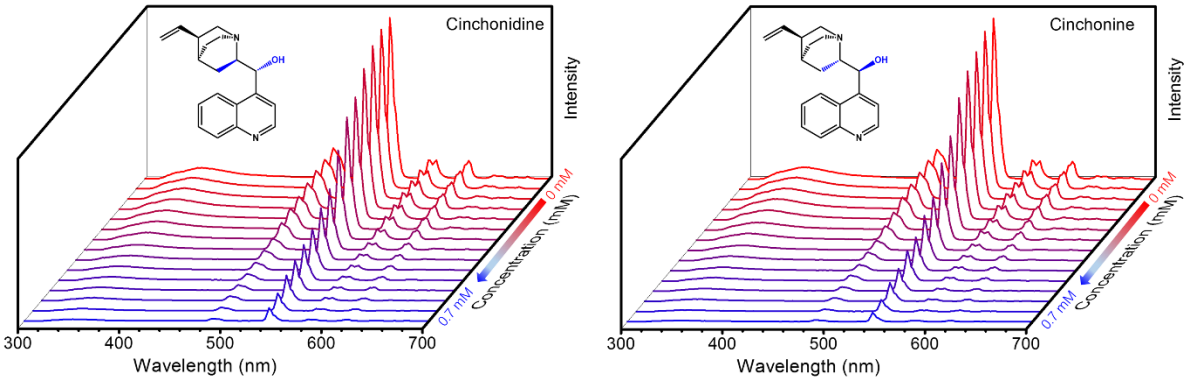
Supplementary Figure 16 Divisions of the intensities of the excitation spectra of Zn-MOF-C-Tb and Zn-MOF-C-Tb with Cinchonine. The solid line is the intensity ratio of Zn-MOF-C-Tb and Zn-MOF-C-Tb with Cinchonine, while the dotted line is the excitation spectra of Zn-MOF-C-Tb. From the lines we can get that Zn-MOF-C-Tb can be effectively excited at ~270-345 nm (effective range, the range of the excitation wavelength which can effectively excite the MOF), while Zn-MOF-C-Tb can be effectively quenched by Cinchonine at ~265-300 nm (sensitive range, the range of the excitation wavelength which can effectively quench the luminescence of the MOF by the analytes). Their intersection (270-300 nm) is the effective and sensitive range for further needs.



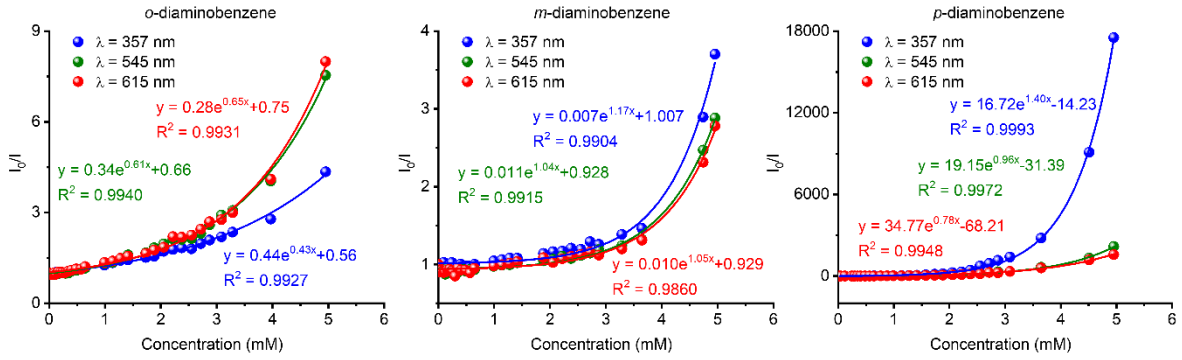
Supplementary Figure 17 Excitation spectra of Zn-MOF-C-Tb with Cinchonine and Cinchonidine at 544 nm. Reprinted with permission from ref. 2, Nature.



Supplementary Figure 18 Divisions of the intensities of the excitation spectra of Zn-MOF-C-Tb with Cinchonine (I_1) and Cinchonidine (I_2). 292 nm is selected for effective, sensitive, and selective needs. The sensitive and effective range is discussed in details in Supplementary Figure 16.



Supplementary Figure 19 Emission spectra of Zn-MOF-C-Tb with the additions of Cinchonine and Cinchonidine. Reprinted with permission from ref. 2, Nature.

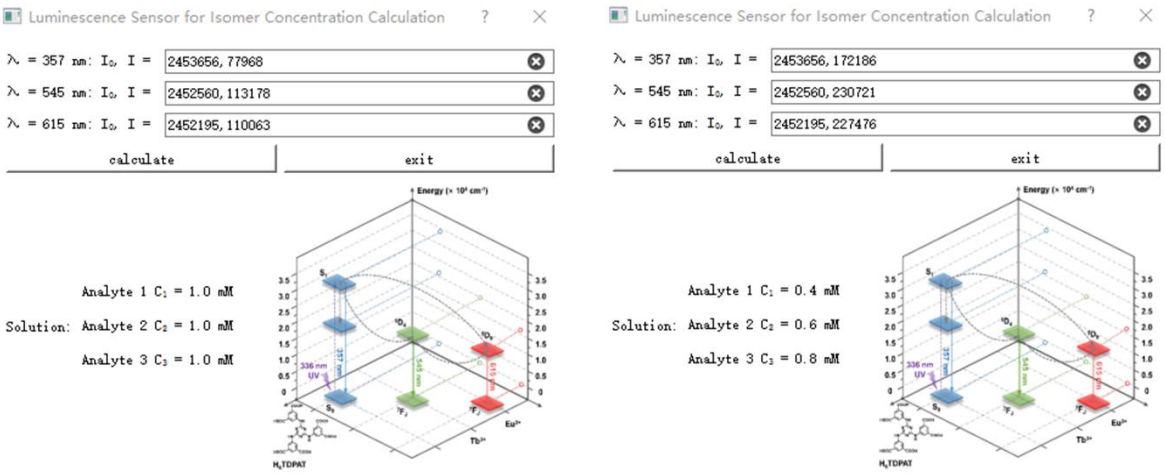


Supplementary Figure 20 Intensities changes of the emission spectra of EuTb@NKU-102 with the additions of phenylenediamine. Reprinted with permission from ref. 3, Chinese Chemical Society.

Supplementary Table 1 Fitting parameters of

$\ln(I_0/I) = a_1C_1^2+a_2C_2^2+a_3C_3^2+a_4C_1C_2+a_5C_1C_3+a_6C_2C_3+a_7C_1+a_8C_2+a_9C_3+a_{10}$.

λ	a_1	a_2	a_3	a_4	a_5	
357	-0.0075	0.0500	-0.1061	0.1818	0.1574	
545	-0.0212	0.0517	-0.1562	0.1852	-0.0160	
615	-0.0021	0.0016	-0.1423	0.0591	0.0871	
λ	a_6	a_7	a_8	a_9	a_{10}	R^2
357	0.3148	0.0127	-0.3110	2.5648	0.5922	0.9987
545	0.2419	0.2129	-0.2647	2.3782	0.4643	0.9974
615	0.1785	0.1960	-0.0183	2.3533	0.3909	0.9957



Supplementary Figure 21 Testing samples for the detection of the mixture of *o*-, *m*- and *p*-phenylenediamine by a calculation software programmed by ourself. The fitting parameters in this program is shown in Supplementary Table 1, while the original codes are listed below. Reprinted with permission from ref. 3, Chinese Chemical Society.

References

1. Han, Z., Wang, K., Min, H., Xu, J., Shi, W., Cheng, P. Bifunctionalized metal-organic frameworks for pore-size-dependent enantioselective sensing. *Angew. Chem. Int. Ed*. 61, e202204066 (2022).

2. Han, Z., Wang, K., Guo, Y., Chen, W., Zhang, J., Zhang, X., Siligardi, G., Yang, S., Zhou, Z., Sun, P., Shi, W., Cheng, P. Cation-induced chirality in a bifunctional metal-organic framework for quantitative enantioselective recognition. *Nat. Commun.* **10**, 5117 (2019).

3. Han, Z., Wang, K., Chen, Y., Li, J., Teat, S.J., Yang, S., Shi, W., Cheng, P. A multicenter metal-organic framework for quantitative detection of multicomponent organic mixtures. *CCS Chem*. **4**, 3238-3245 (2022).

Supplementary Software (by Python)

```
from PyQt5.QtCore import (PYQT_VERSION_STR,Qt)

from PyQt5.QtWidgets import (QApplication, QDialog, QHBoxLayout,
QLabel,QLineEdit,QMessageBox, QPushButton,QVBoxLayout)

from PyQt5.QtCore import *

from PyQt5.QtWidgets import *

from PyQt5.QtGui import QPixmap

import math

import apprcc_rc


GRID = 100

C1_MIN,C1_MAX = 0,5

C2_MIN,C2_MAX = 0,5

C3_MIN,C3_MAX = 0,5

DATA_CNT = 3

CANDIDATE_CNT = 1


def calc(lambda_,C1,C2,C3):

   if lambda_ == 357:

      a1,a2,a3,a4,a5,a6,a7,a8,a9,a10 = -0.0075,0.0500,-0.1061,0.1818,0.1574,0.3148,0.0127,-
0.3110,2.5648,0.5922

   if lambda_ == 545:

      a1,a2,a3,a4,a5,a6,a7,a8,a9,a10 = -0.0212,0.0517,-0.1562,0.1852,-0.0160,0.2419,0.2129,-
0.2647,2.3782,0.4643

   if lambda_ == 615:

      a1,a2,a3,a4,a5,a6,a7,a8,a9,a10 = -0.0021,0.0016,-0.1423,0.0591,0.0871,0.1785,0.1960,-
0.0183,2.3533,0.3909

   a = a1*C1**2 + a2*C2**2 + a3*C3**2 + a4*C1*C2 + a5*C1*C3 + a6*C2*C3 + a7*C1 + a8*C2 +
a9*C3 + a10

   return round(a, 3)


def solve(lambda_,I0,I):

   y = [math.log(I0[0]/I[0]), math.log(I0[1]/I[1]), math.log(I0[2]/I[2])]

   distance = [float('inf')]*CANDIDATE_CNT
```

```
    C=[(-1,-1,-1)]*CANDIDATE_CNT
    for i in range(1+GRID):
        for j in range(1+GRID):
            for k in range(1+GRID):
                C1_tmp,C2_tmp,C3_tmp = C1_MIN+(C1_MAX-C1_MIN)*i/GRID, C2_MIN+(C2_MAX-
C2_MIN)*j/GRID, C3_MIN+(C3_MAX-C3_MIN)*k/GRID
                y_ = [calc(lambda_[0],C1_tmp,C2_tmp,C3_tmp), calc(lambda_[1],C1_tmp,C2_tmp,C3_tmp),
calc(lambda_[2],C1_tmp,C2_tmp,C3_tmp)]
                curr_distance = (y[0]-y_[0])**2 + (y[1]-y_[1])**2 +(y[2]-y_[2])**2
                max_loss = max(distance)
                l = distance.index(max_loss)
                if distance[l] > curr_distance:
                    distance[l] = curr_distance
                    C[l] = C1_tmp, C2_tmp, C3_tmp
    return  C,[(calc(lambda_[0],C[i][0],C[i][1],C[i][2]), calc(lambda_[1],C[i][0],C[i][1],C[i][2]),
calc(lambda_[2],C[i][0],C[i][1],C[i][2])) for i in range(CANDIDATE_CNT)], distance


class MainForm(QDialog):
    def __init__(self):
        super(MainForm, self).__init__()


        label_data1=QLabel("<html>&lambda;</html> = 357 nm: I<sub>0</sub>, I = ")
        self.textbox_data1 = QLineEdit()
        self.textbox_data1.setPlaceholderText("")
        self.textbox_data1.setToolTip("eee")
        self.textbox_data1.setClearButtonEnabled(True)


        label_data2=QLabel("<html>&lambda;</html> = 545 nm: I<sub>0</sub>, I = ")
        self.textbox_data2 = QLineEdit()
        self.textbox_data2.setPlaceholderText("")
        self.textbox_data2.setClearButtonEnabled(True)


        label_data3=QLabel("<html>&lambda;</html> = 615 nm: I<sub>0</sub>, I = ")
        self.textbox_data3 = QLineEdit()
```

```
self.textbox_data3.setPlaceholderText("")
self.textbox_data3.setClearButtonEnabled(True)

calcButton = QPushButton("calculate")
quitButton = QPushButton("exit")

dataLayout = QVBoxLayout()

dataLayout1 = QHBoxLayout()
dataLayout1.addWidget(label_data1)
dataLayout1.addWidget(self.textbox_data1)

dataLayout2 = QHBoxLayout()
dataLayout2.addWidget(label_data2)
dataLayout2.addWidget(self.textbox_data2)

dataLayout3 = QHBoxLayout()
dataLayout3.addWidget(label_data3)
dataLayout3.addWidget(self.textbox_data3)

dataLayout.addLayout(dataLayout1)
dataLayout.addLayout(dataLayout2)
dataLayout.addLayout(dataLayout3)

buttonLayout = QHBoxLayout()
buttonLayout.addWidget(calcButton)
buttonLayout.addWidget(quitButton)

resultLayout = QHBoxLayout()
resultLayout1 = QLabel("Solution: ")
resultLayout1.setFixedWidth(56)
self.resultLayout2 = QLabel(" ")
```

```
self.resultLayout2.setFixedWidth(142)
self.resultLayout2.setAlignment(Qt.AlignLeft)
self.resultLayout2.setAlignment(Qt.AlignVCenter)

label_image = QLabel()
label_image.setScaledContents(True)
image1 = QPixmap(":/image/bg.tif")
label_image.setPixmap(image1)
label_image.setFixedSize(QSize(240,240))

vbox=QVBoxLayout()
vbox.addWidget(label_image)

resultLayout.addWidget(resultLayout1)
resultLayout.addWidget(self.resultLayout2)
resultLayout.addLayout(vbox)

layout = QVBoxLayout()
layout.addLayout(dataLayout, 1)
layout.addLayout(buttonLayout)
layout.addLayout(resultLayout)
self.setLayout(layout)

self.textbox_data1.editingFinished.connect(self.sanitizer)
self.textbox_data2.editingFinished.connect(self.sanitizer)
self.textbox_data3.editingFinished.connect(self.sanitizer)

quitButton.clicked.connect(self.done)
calcButton.clicked.connect(self.calc)

self.setMinimumWidth(470)
self.setMinimumHeight(240)
```

```
        self.setWindowTitle("Luminescence Sensor for Isomer Concentration Calculation")


    def sanitizer(self):
        self.lambda_, self.I0, self.I = [357,545,615],[],[]
        data=[self.textbox_data1.text().replace("，",","), self.textbox_data2.text().replace("，",","), self.textbox_data3.text().replace("，",",")]
        for d in data:
            if len(d.split(','))==1 and len(d)!=0:
                message=QMessageBox(QMessageBox.NoIcon, "Exception", "Input data  incomplete, please check")
                message.exec()
                return False
            if len(d.split(',')) ==2:
                try:
                    I0,I = float(d.split(',')[0]), float(d.split(',')[1])
                except Exception as e:
                    message=QMessageBox(QMessageBox.NoIcon, "Exception", "Please ensure all parameters are numbers")
                    message.exec()
                    return False
                self.I0.append(I0)
                self.I.append(I)
        if len(self.I) != 3:
            return False
        return True


    def calc(self):
        if not self.sanitizer():
            return False
        C,Y,all_distance = solve(self.lambda_,self.I0,self.I)
        dis_idx = {}
        for idx,distance in enumerate(all_distance):
            dis_idx[distance] = idx
```

```
        sort_distance = sorted(all_distance)
        for distance in sort_distance:
            idx = dis_idx[distance]
            self.resultLayout2.setText("Analyte 1  C<sub>1</sub> = {C1} mM<br><br>Analyte 2 C<sub>2</sub> = {C2} mM <br><br>Analyte 3  C<sub>3</sub> = {C3} mM".format(C1= C[idx][0],C2= C[idx][1],C3= C[idx][2],y_=Y[idx], d=round(distance,3)))
        return
    def event(self, event):
        if event.type()==QEvent.EnterWhatsThisMode:
            QWhatsThis.leaveWhatsThisMode()
            self.help()
        return QDialog.event(self,event)

    def help(self):
        message=QMessageBox(QMessageBox.NoIcon, "About", "Version: 1.0")
        message.exec()

if __name__ == '__main__':

    import sys
    app = QApplication(sys.argv)
    form = MainForm()
    form.setStyleSheet("background-color:rgb(255,255,255)")
    form.show()
    app.exec_()
    del form
```