Electronic Supplementary Information

Single cell imaging reveals cisplatin regulating interactions between transcription (co)factors and DNA

Yu Lin,‡[a] Kui Wu,‡[b] Feifei Jia,‡[a] Ling Chen,[a] Zhaoying Wang,[a] Yanyan Zhang,[a] Qun Luo,[ac] Suyan Liu,[a] Luyu Qi,[ac] Nan Li,[cd] Pu Dong,[e] Fei Gao,[e] Wei Zheng,[a] Xiaohong Fang,*[cd] Yao Zhao*[a] and Fuyi Wang*[acf]

[a] Beijing National Laboratory for Molecular Sciences; CAS Research/Education Center for Excellence in Molecular Sciences; National Centre for Mass Spectrometry in Beijing, CAS Key Laboratory of Analytical Chemistry for Living Biosystems, Institute of Chemistry, Chinese Academy of Sciences, Beijing, 100190, People's Republic of China

[b.] Key Laboratory of Hubei Province for Coal Conversion and New Carbon Materials; School of Chemistry and Chemical Engineering, Wuhan University of Science and Technology, Wuhan, 430081, People's Republic of China

[c] University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

[d] Beijing National Laboratory for Molecular Sciences; CAS Research/Education Center for Excellence in Molecular Sciences; Key Laboratory of Molecular Nanostructures and Nanotechnology, Institute of Chemistry, Chinese Academy of Sciences, Beijing 100190, P.R. China

[e] China Telecom Corporation Limited Beijing Research Institute, Beijing, 100035, People's Republic of China

[f] College of Traditional Chinese Medicine, Shandong University of Traditional Chinese Medicine, Jinan 250355, People's Republic of China

‡ The authors contributed equally to this work.

Email: yaozhao@iccas.ac.cn (Y.Zhao); fuyi.wang@iccas.ac.cn (F.W.); xfang@iccas.ac.cn (X. F.)

Supplementary Figures

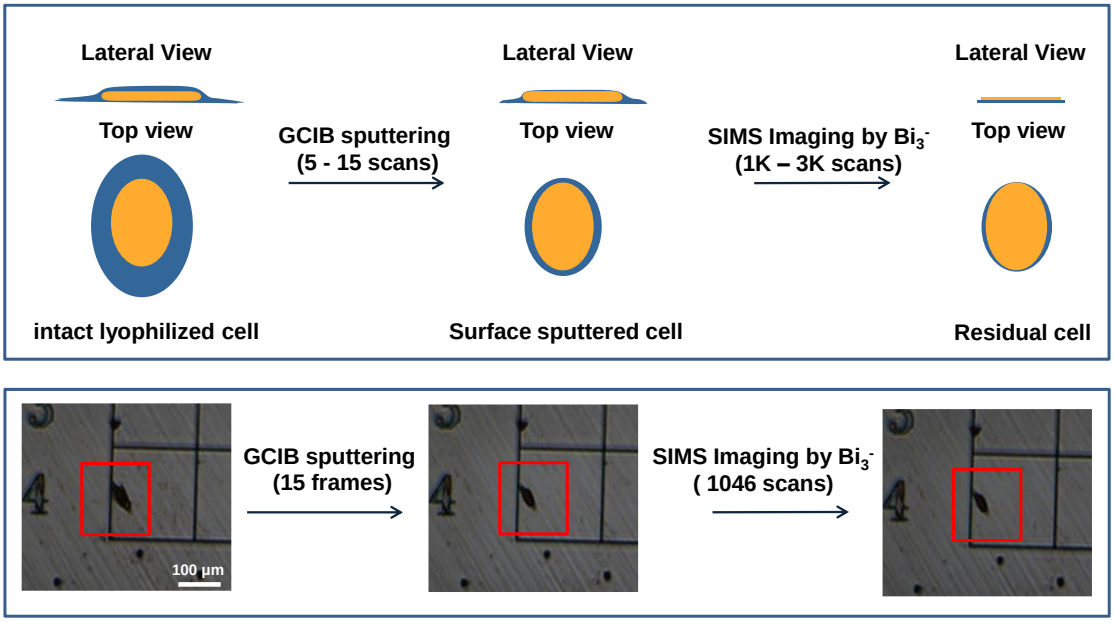


Fig. S1. Workflow of a typical ToF-SIMS imaging of a single cell. Yellow circles and blue circles represent the cell nuclear and the membrane/cytoplasm areas, respectively. Bottom panel: optical views of a lyophilized cell in the red box were scanned for 1046 times by using Bi_3^+ ion beam.

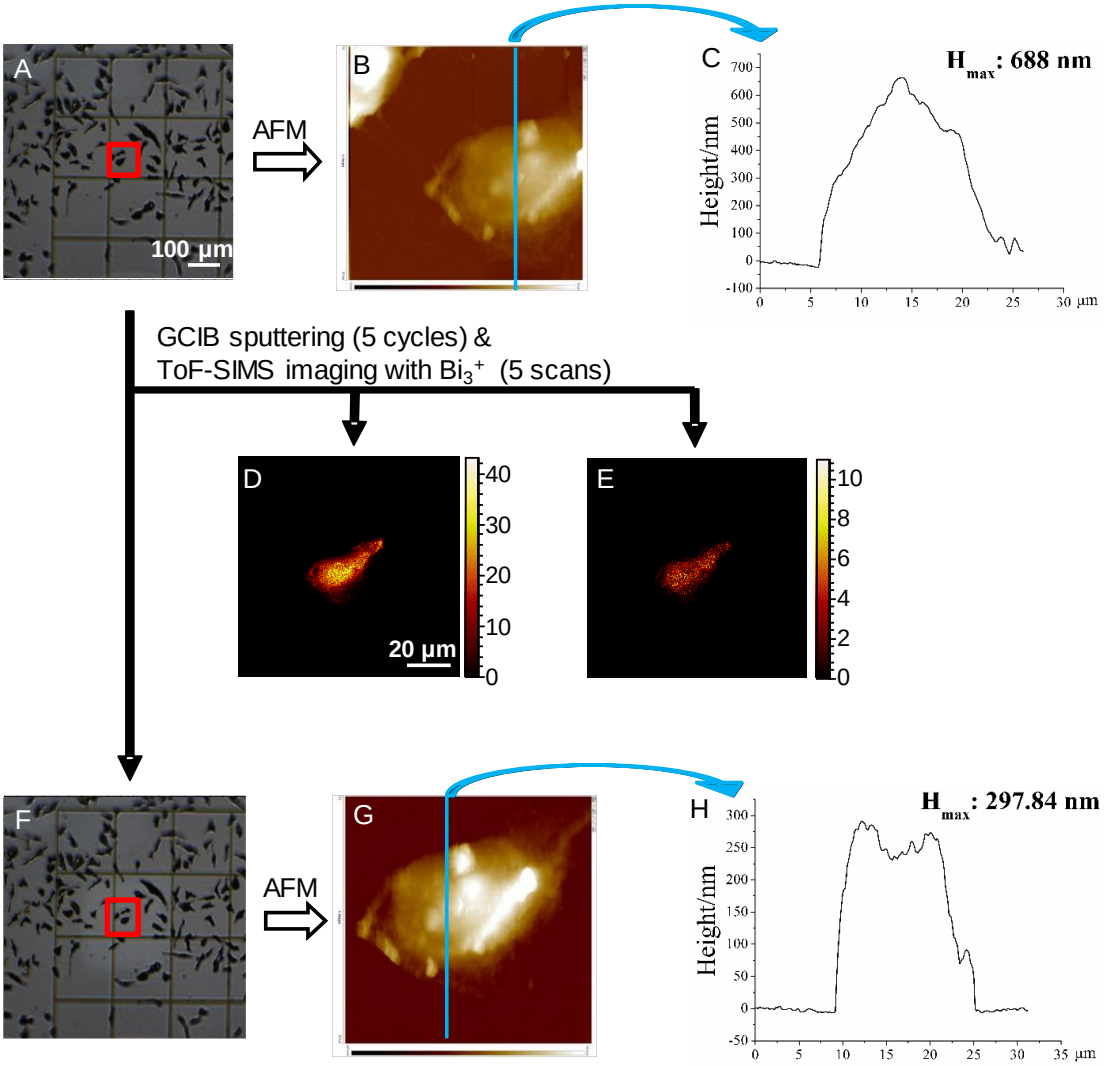


Fig. S2. ToF-SIMS images and AFM morphological images of single cells. AFM imaging (B, C) was first performed on the lyophilized single cells in the red box shown in the bright field image A. The same cells were then sputtered by 5 cycles of Argon gas cluster ion beam (GCIB) following by 5 times of ToF-SIMS scanning with Bi_3^+ ion beam to obtain images of total ions and PO_3^- as shown in D and E. Finally, AFM image (G, H) was recorded on the same cells (F). The results indicate that 5 cycles of GCIB sputtering was sufficient for removal of the membrane for this cell.

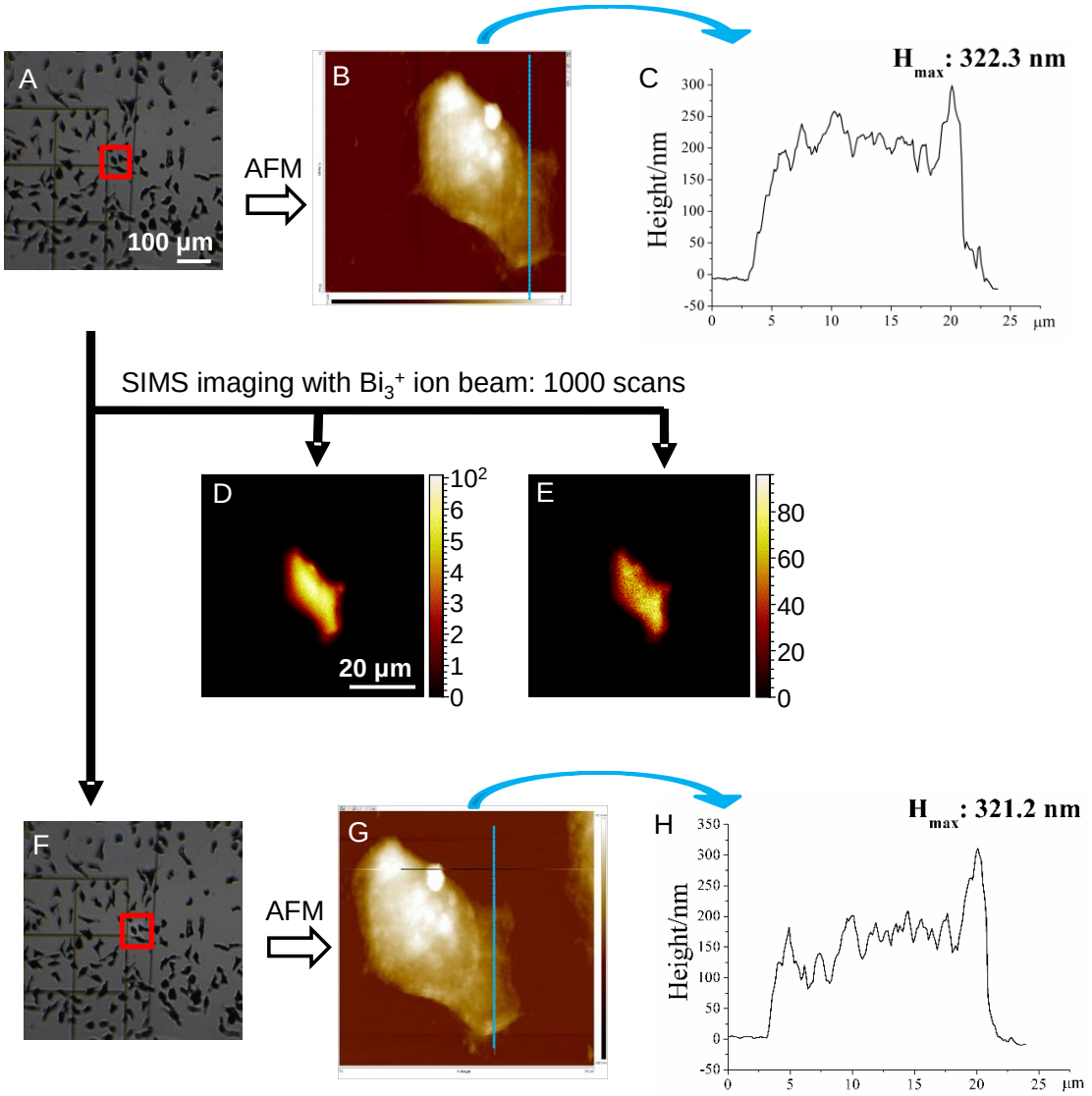


Fig. S3. ToF-SIMS images and AFM morphological image of single cells. AFM imaging (B, C) was first performed on the lyophilized single cells in the red box of the bright field image A. Then, ToF-SIMS images were recorded on the same cell by 1000 scans with Bi_3^+ ion beam (D and E). Finally, AFM image (G, H) was again recorded on the same cells (F). The results indicate that 1000 Bi_3^+ ion beam scans did not significantly change the cell morphology.

Ions (theoretic *m/z*)	$[^{194}PtCN]^-$ (219.97)	$[^{195}PtCN]^-$ (220.97)	$[^{196}PtCN]^-$ (221.98)
Relative Isotopic abundance (%)	35.44	36.89	27.67
Peak area detected by SIMS	5891	6115	4488
Relative peak area (%)	35.72	37.07	27.20

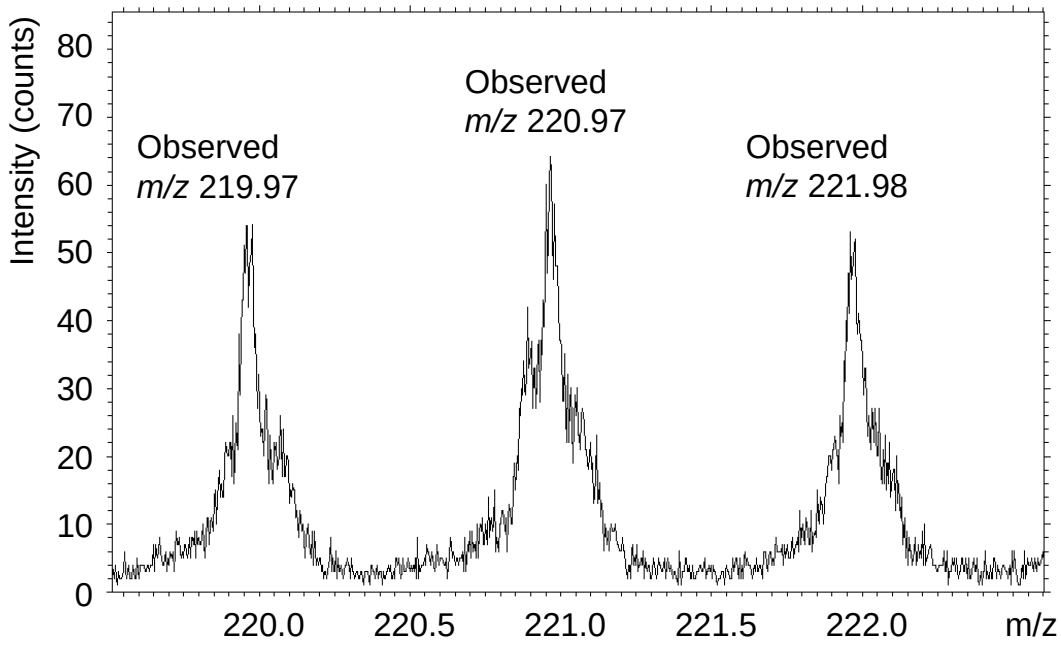


Fig. S4. ToF-SIMS spectrum of $[PtCN]^-$ ion produced in HeLa cells treated with 50 μM cisplatin for 24 hours. The data listed in the table are theoretic *m/z*, relative isotopic abundance, and peak areas and relative peak areas of $[PtCN]^-$ ions detected by ToF-SIMS.

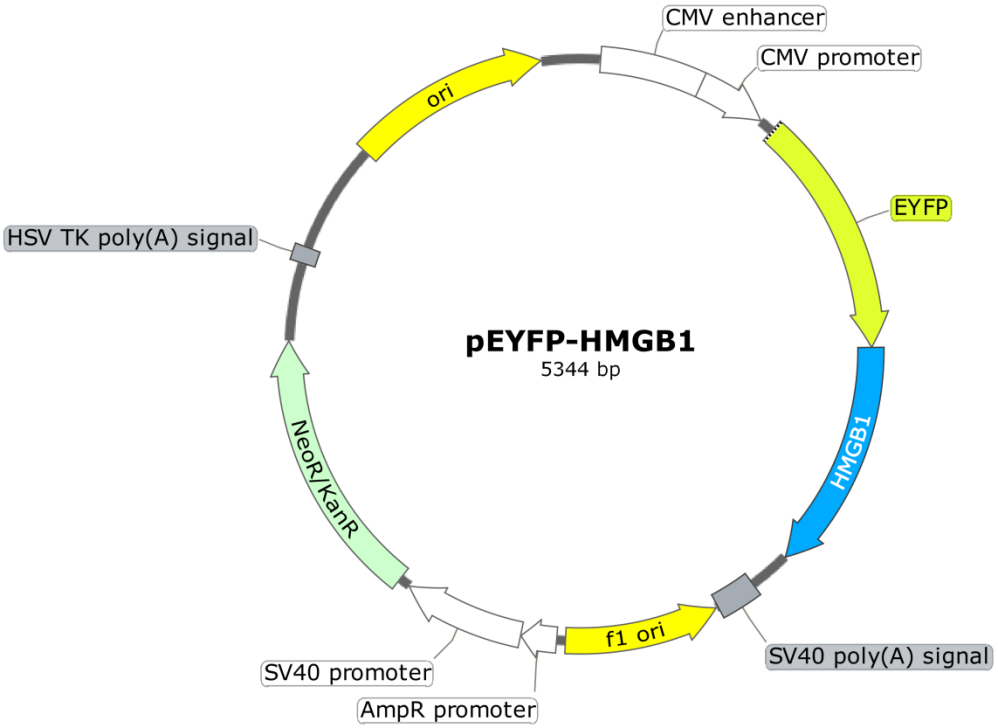
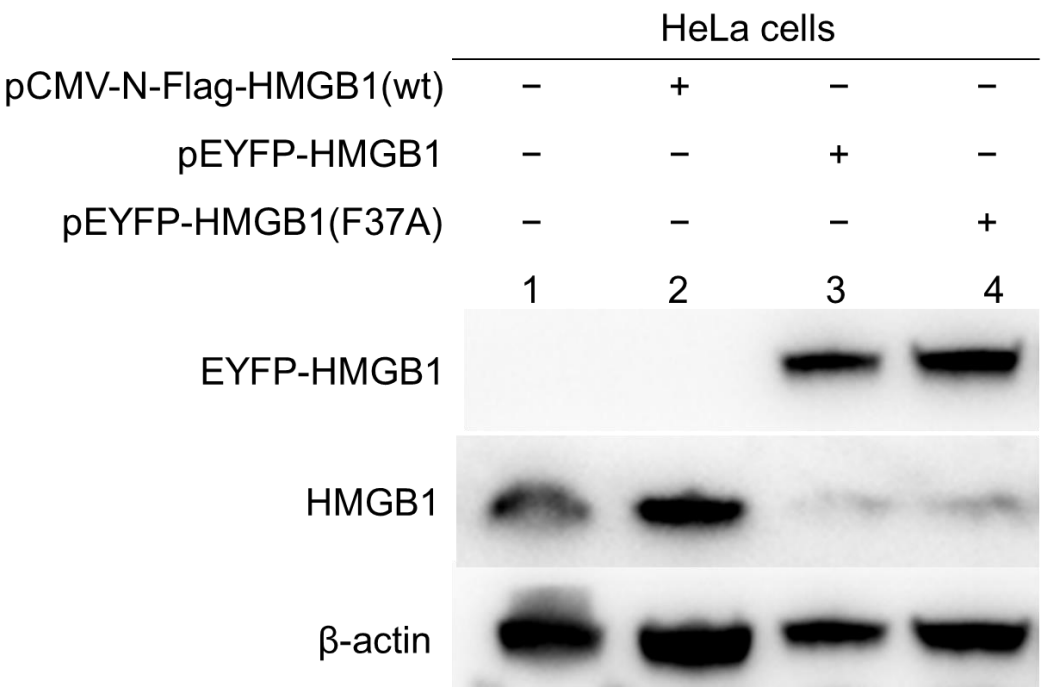


Fig. S5. Schematic diagram of pEYFP-HMGB1 expression vector. The full length of HMGB1 coding sequence was inserted between HindIII and BamHI restriction enzymatic digest sites of the multiple clone site (MCS).



	Signal intensity or ratio			
Lanes	1	2	3	4
EYFP-HMGB1	–	–	2142	2760(F37A)
HMGB1	1546(wt)	2255(wt)	272(wt)	494(wt)
β-actin	3072	3348	1788	2287
HMGB1/β-actin	0.503	0.674	0.152	0.216
EYFP-HMGB/β-actin	–	–	1.198	1.207

Fig. S6. Western blotting results of the proteins extracted from HeLa cells transfected by various exogenous plasmids. Lane 1: HeLa cells without exogenous plasmid transfection; Lane 2, 3, 4: HeLa Cells transfected by pCMV-N-Flag-HMGB1(wt), pEYFP-HMGB1, and pEYFP-HMGB1(F37A) plasmids, respectively. The results showed the successful expression of the wild type, EYFP-fused wild type HMGB1 and EYFP-fused mutant HMGB1(F37A) in HeLa cells. The density values of each WB band are listed in the table below the WB graphics.

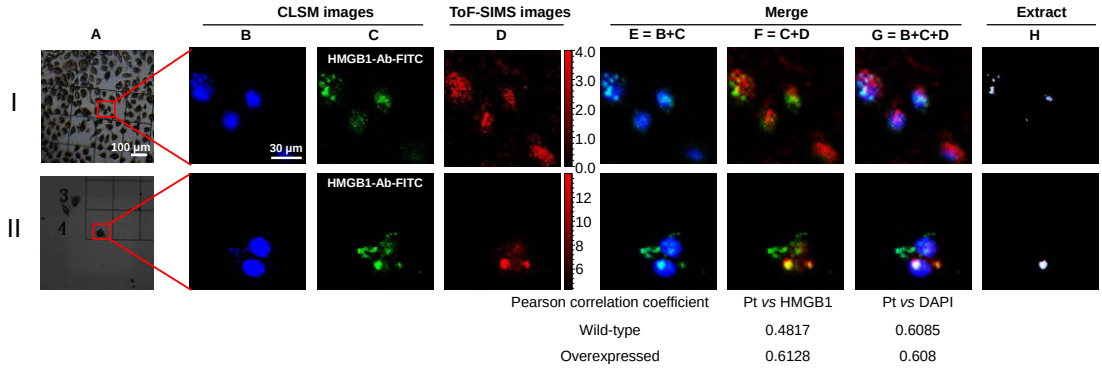


Fig. S7. COSIMSi of HeLa cells (I) without exogenous plasmid transfection and (II) transfected by pEYFP-HMGB1(wt) plasmid, and treated with 50 µM cisplatin. (A) Bright field images; (B) Fluorescence images of DAPI (λ_{ex} = 405 nm; λ_{em} = 425 – 475 nm); (C) fluorescence images of immunostained HMGB1 (λ_{ex} = 488 nm, λ_{em} = 500 – 600 nm); (D) ToF-SIMS image of $[PtCN]^-$ acquired at *m/z* 220, 221 and 222; (E) Merged fluorescent image of DAPI (blue) and immunostained HMGB1 (green); (F) Merged image of immunostained HMGB1 (green) and SIMS image of $[PtCN]^-$ (red); (G) Merged images of fluorescence images of DAPI, immunostained HMGB1 and SIMS image of $[PtCN]^-$; (H) Extracted images from (G) for better contrast.

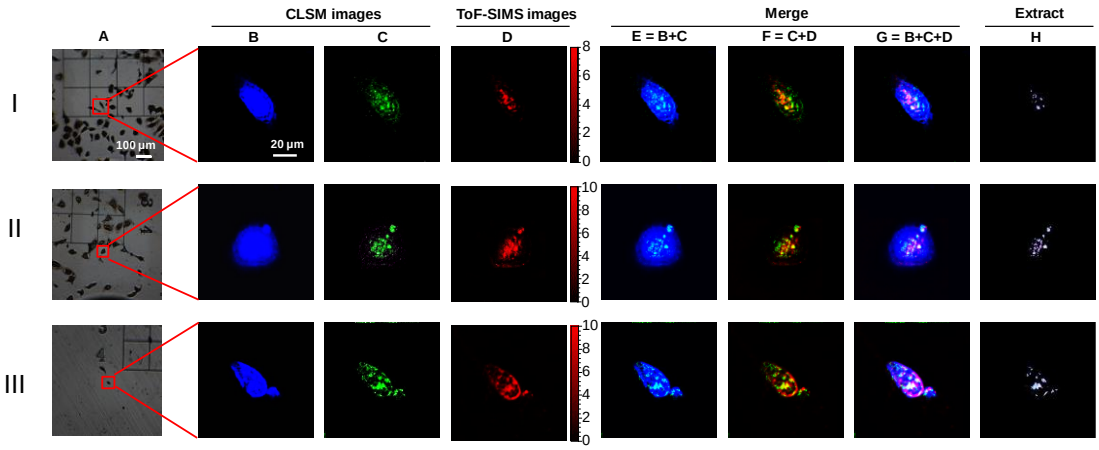


Fig. S8. COSIMSi of HeLa cells transfected by pEYFP-HMGB1(wt) plasmid and treated with different concentration of cisplatin. (I) 25 μM; (II) 50 μM; (III) 100 μM. (A) Bright field images; (B) Fluorescence images of DAPI (λ_{ex} = 405 nm; λ_{em} = 425 – 475 nm); (C) Fluorescence images of EYFP-HMGB1 (λ_{ex} = 488 nm; λ_{em} = 500 – 600 nm); (D) ToF-SIMS images of $[PtCN]^-$ acquired at *m/z* 220, 221 and 222; (E) Merged fluorescent images of DAPI (blue) and EYFP-HMGB1(green); (F) Merged images of EYFP-HMGB1 (green) and SIMS image of $[PtCN]^-$ (red); (G) Merged images of fluorescence images of DAPI, HMGB1 and SIMS image of $[PtCN]^-$; (H) Extracted images from (G) for better contrast.

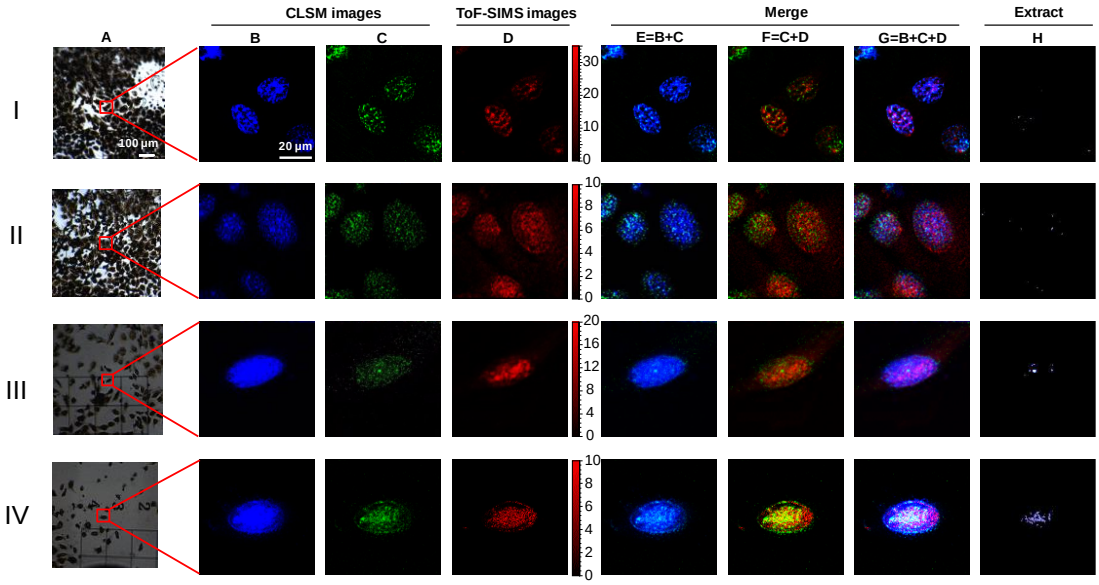


Fig. S9. COSIMSi of HeLa cells transfected by pEYFP-HMGB1(wt) plasmid and treated with 50 μM cisplatin for different time. (I) 3 h; (II) 6 h; (III) 12 h; (IV) 24 h. (A) Bright field images; (B) Fluorescence images of DAPI (λ_{ex} = 405 nm; λ_{em} = 425 – 475 nm); (C) Fluorescence images of EYFP-HMGB1 (λ_{ex} = 488 nm; λ_{em} = 500 – 600 nm); (D) ToF-SIMS images of $[PtCN]^-$ acquired at *m/z* 220, 221 and 222; (E) Merged fluorescent images of DAPI (blue) and EYFP-HMGB1 (green); (F) Merged images of EYFP-HMGB1 (green) and SIMS image of $[PtCN]^-$ (red); (G) Merged images of fluorescence images of DAPI, HMGB1 and SIMS image of $[PtCN]^-$; (H) Extracted images from (G) for better contrast.

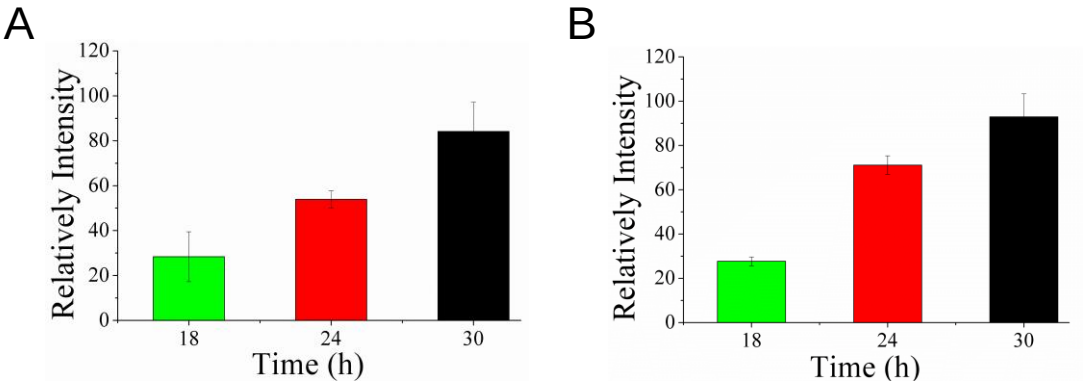


Fig. S10. SMAD transcription activity level of HeLa cells (A) without and (B) with transfection with exogenous Smad3 plasmid, cultured in FBS containing DMEM for different times. The cells were cultured under serum-starved model for 12, 6 and 0 h, respectively, after transfected with SMAD luciferase reporter plasmid and Renilla luciferase reporter plasmid. The culture medium was then replaced to FBS-containing DMEM medium and grew for further 18, 24 and 30 h, respectively. The relative intensity of illumination of SMAD-Luc reporter to Renilla-Luc reporter increased with increase in the culturing time in FBS-containing medium, indicating that the FBS we used in this work contained TGF-β that stimulated the Smad proteins to enter the nucleus and induce expression of luciferase reporters.

Supplementary Code

Java Code for Extraction of Merged Images

```
import java.awt.Color;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;

public class Merge
{
   public static void main(String[] args)
   {
      String str1 = "";
      String str2 = "";
      String str3 = "";
      int i = 0;

      if (args.length == 4) {
         str1 = args[0];
         str2 = args[1];
         str3 = args[2];
         i = Integer.parseInt(args[3]);
      }
      else if (args.length == 3) {
         str1 = args[0];
         str2 = args[1];
         str3 = args[2];
         i = 100;
      }
      else if (args.length == 1) {
         str1 = "DAPI.jpg";
         str2 = "HMGB1.jpg";
         str3 = "Pt.jpg";
         i = Integer.parseInt(args[0]);
      }
      else {
         str1 = "DAPI.jpg";
         str2 = "HMGB1.jpg";
         str3 = "Pt.jpg";
         i = 100;
      }

      BufferedImage localBufferedImage1 = readImage(str1);
```

```
    BufferedImage localBufferedImage2 = readImage(str2);
    BufferedImage localBufferedImage3 = readImage(str3);

    int[][] arrayOfInt1 = convertImageToArray(localBufferedImage1, 3);
    int[][] arrayOfInt2 = convertImageToArray(localBufferedImage2, 2);
    int[][] arrayOfInt3 = convertImageToArray(localBufferedImage3, 1);

    int j = localBufferedImage1.getWidth();
    int k = localBufferedImage1.getHeight();

    int[][] arrayOfInt4 = new int[j][k];

    for (int m = 0; m < j; m++) {
      for (int n = 0; n < k; n++)
      {
        Color localColor;
        if ((arrayOfInt3[m][n] > i) && (arrayOfInt2[m][n] > i) && (arrayOfInt1[m][n] > i)) {
          localColor = new Color(arrayOfInt3[m][n], arrayOfInt2[m][n], arrayOfInt1[m][n]);
          arrayOfInt4[m][n] = localColor.getRGB();
        }
        else {
          localColor = new Color(0, 0, 0);
          arrayOfInt4[m][n] = localColor.getRGB();
        }

      }

    }

    writeImageFromArray("result.jpg", "jpg", arrayOfInt4);
  }

  public static BufferedImage readImage(String paramString)
  {
    File localFile = new File(paramString);
    BufferedImage localBufferedImage = null;
    try {
      localBufferedImage = ImageIO.read(localFile);
    } catch (IOException localIOException) {
      localIOException.printStackTrace();
    }

    return localBufferedImage;
  }
```

```
    public static int[][] convertImageToArray(BufferedImage paramBufferedImage, int paramInt)
    {
        int i = paramBufferedImage.getWidth();
        int j = paramBufferedImage.getHeight();

        int[] arrayOfInt = new int[i * j];
        paramBufferedImage.getRGB(0, 0, i, j, arrayOfInt, 0, i);

        int[][] arrayOfInt1 = new int[j][i];

        for (int k = 0; k < j; k++) {
            for (int m = 0; m < i; m++) {
                Color localColor = new Color(arrayOfInt[(k * i + m)]);

                if (paramInt == 1) {
                    arrayOfInt1[k][m] = localColor.getRed();
                }
                else if (paramInt == 2) {
                    arrayOfInt1[k][m] = localColor.getGreen();
                }
                else {
                    arrayOfInt1[k][m] = localColor.getBlue();
                }
            }

        }

        return arrayOfInt1;
    }

    public static void writeImageFromArray(String paramString1, String paramString2, int[][]
paramArrayOfInt)
    {
        int i = paramArrayOfInt[0].length;
        int j = paramArrayOfInt.length;

        int[] arrayOfInt = new int[i * j];
        for (int k = 0; k < j; k++) {
            for (int m = 0; m < i; m++) {
                arrayOfInt[(k * i + m)] = paramArrayOfInt[k][m];
            }
        }
```

```
        BufferedImage localBufferedImage = new BufferedImage(i, j, 4);
        localBufferedImage.setRGB(0, 0, i, j, arrayOfInt, 0, i);
        try
        {
          File localFile = new File(paramString1);
          ImageIO.write(localBufferedImage, paramString2, localFile);
        } catch (IOException localIOException) {
          localIOException.printStackTrace();
        }
      }
    }
```