

Electronic Supplementary Information (ESI) for “DNA  
Origami-Engineered Gold Nanoparticle Multimers for  
Ultrasensitive, Label-Free SERS Detection of Small  
Molecules and Biomolecules”

Yaosen Wang,<sup>a</sup> Yanlong Cai,<sup>b</sup> Huaizhou Jin<sup>b\*</sup> and Shangzhong Jin<sup>a\*</sup>

<sup>a</sup>College of Optical and Electronic Technology, China Jiliang University, Hangzhou 310018,  
China.

<sup>b</sup>Key Laboratory of Quantum Precision Measurement, College of Physics, Zhejiang University  
of Technology, Hangzhou, China.

\* Corresponding author: jinsz@cjl.u.edu.cn;

## 1. Table of strands

|               |  |
|---------------|--|
| Core staple1  | AAGGGACTGGCGCGTCGCCGCGAGCGAGAAAAGGA                |
| Core staple2  | TGACCTCCGCTACAGTAGCCGGTCAAGA                       |
| Core staple3  | TTCTGCGAAGAAAAATAACAGAGGTGAGGCCATTACATTG           |
| Core staple4  | CTACAATAATGAGTGACCCTGATTAGAGCCCGTAAT               |
| Core staple5  | TTCATTGCAATACTTTTAGCGAAAATCACC                     |
| Core staple6  | AACCGGCCTCAGGAAGATCCGTTGTAA                        |
| Core staple7  | ATAGAAACCAGGCAAAGCGAAAGGGGG                        |
| Core staple8  | GTCATAGAATGGGATAGGTCACGTTGGTCGTAATCA               |
| Core staple9  | AGCCCCGATAACAGTATGAGCAATTTTATTTTC                  |
| Core staple10 | GTTAGCGTAACGACGTTGAAA                              |
| Core staple11 | AACAAGCAAGCCAGAGCCT                                |
| Core staple12 | GGAACGGTACGTCTGTCGATTGTTCCAGAAG                    |
| Core staple13 | ATAAGAGCAAGAAAAATAAATGTACCGACAGTC                  |
| Core staple14 | TTATCAACAATATTGTTTA                                |
| Core staple15 | TTAGTACCGACTTTATACATTTACGACCAGTAATATACCGAACCTCAA   |
| Core staple16 | AACGACGAGCGCTGCGCAGCGAAATTGACGGTAAA                |
| Core staple17 | ATTAATGAATCCGGAATAAATACCCATCCTAATT                 |
| Core staple18 | GAACAACGTATCGGAGGGAGTGAGGCT                        |
| Core staple19 | GCTGCAAGCGACGTGCTTTC                               |
| Core staple20 | GCAAAAAGAAATACCCAAACAAGAGGATTTAGAAGTTGCTGAACGAACCA |
| Core staple21 | AGCTGGCGCCAAGTGTTTTT                               |
| Core staple22 | GCAAATCAGATATCTTACC                                |
| Core staple23 | AGAATAACATAAAAATTCTGTCACAACATCACTGCC               |
| Core staple24 | ATAATAACTCCAAAGCGGGATACGAGGGTAGCA                  |
| Core staple25 | GAGCGGACGAACGTACCGCAAACCCTTCTGACCTTTAATGCAATCAAC   |
| Core staple26 | CGCATAGGGTCAATGTATTAACCAGGC                        |
| Core staple27 | CGAGTGACACGGATATATTAGTAATCATTTCATTATAACAACATTTA    |
| Core staple28 | AACCTCCCGACTTTGCACC                                |
| Core staple29 | AGCGAACGGCCGGACCGGTTGAAATTTCG                      |
| Core staple30 | AACATCACAATATTTATTAATCTAACAATAAGGTTGGCAGCGAACT     |
| Core staple31 | CCCGCCACCACATAATCTCAGACTGACAGAAGAGCCATGGCTTAATTCG  |
| Core staple32 | CATTAAATTTTTGTCAATCATAACGGAATGGTGGCA               |
| Core staple33 | TCCATCAAACGGAGAGCTGGAGCTAATTCGCGTCTGG              |
| Core staple34 | GGTAACGCCGGCGCTAGGGC                               |
| Core staple35 | AAAATAATCATAGCTTCCTTAGAAAAACCGCTAA                 |
| Core staple36 | TATTAACCAAGCAGTTAC                                 |
| Core staple37 | TAACCTTATTAACAAAGCTGCTCTCATTACTTTAACGTATTTGGA      |
| Core staple38 | CTTAAACCATTTCAACAGTTTCAGCGGATTCGAGGTATTCGGTTTTTCA  |
| Core staple39 | CGCCACCCTCAGACTAAAGTAACAACCTACCGAT                 |
| Core staple40 | AAGAAGTGCAACACCGCCTCCCTCAGATTATTAGCTCGG            |

|               |   |
|---------------|---|
| Core staple41 | GAAAGGGGACGACGACAGTTGCCAAGC                       |
| Core staple42 | TATCAAATACATTTTTCGACAATACTCAGGAGGT                |
| Core staple43 | TAGGAACGCTGAATGGCTTGCTTTATCAGCTTGCTGTGAGAATTCTGTA |
| Core staple44 | CCAGCAGCCAGTCATGACGCTCTAAACTACCAAGTCGCCTGGAAC     |
| Core staple45 | CCAGTATATCACCAAAAATCAAGAAAACGAGAATGTCCACGCCATTCAA |
| Core staple46 | AGAAGAGCGTCGCTGGCCACGGGTCTGA                      |
| Core staple47 | GCTCATTTTTAACGTAATCGGCATGATCGTAGAA                |
| Core staple48 | ATAAGGGCCGAAACTTAGTTAGTTCAGGAAAATC                |
| Core staple49 | CCTCAGAACCACCAGACTAACGGACCAGTC                    |
| Core staple50 | GAAAAACAAATCAAGGTAAATTTATACATTTAACAAAAAGGTTGAGG   |
| Core staple51 | GCATCTATCACACTATTAGCCCGAATG                       |
| Core staple52 | ACGTCAAAAATGAACGCGCCT                             |
| Core staple53 | GAAAAATAATATTAAGAAA                               |
| Core staple54 | AGCTATAATGAATTTAGAAAAG                            |
| Core staple55 | GAATTGAGTTAAGAGAACTGTAAACTATGATAT                 |
| Core staple56 | TTAATATTTTGTTAATAATCAAACCGAGCGCAAAG               |
| Core staple57 | CTGGTCAATTAGAGCTTTGCCATTATCAATCAATATATTAGGATTAGC  |
| Core staple58 | CGGGAGAAGCCTTTATTTCAAACCTCAACAGGTCAGGATTA         |
| Core staple59 | ATACAGGTGCCTTGCAAGAACGACCAGG                      |
| Core staple60 | ATATCAGAGAGATACCTTATTAGAATCGTGATAAA               |
| Core staple61 | TAGGAATCATTAAACGAGCG                              |
| Core staple62 | TCAACCGAGACTTCTATTGACGAAATTTAGCAAATAAGACTACCCACAA |
| Core staple63 | GCTTTAAATGTTTAGTTTTAGTTTTGCC                      |
| Core staple64 | CGGCCTTGCTTAACATTTCTAAAACGGGAGGTTT                |
| Core staple65 | ACGGCTACATAAAGGCTAGCATTCTAATTCCATGTACCGTAAC       |
| Core staple66 | TGAGAAAATTTTAAACAAAAGGGCGACAAAAGAAAGAAACGCCAATGAA |
| Core staple67 | TCAAGAAACAAAATCCACACCACTATGG                      |
| Core staple68 | ATCAAACGCCGCCCTCTACGTTAATATAATGCAAGTAAGATTTGCC    |
| Core staple69 | ATGCCTGCTGATCGGCAAAA                              |
| Core staple70 | CTGTTTGGATAGGGCGGCTTAGACGCTG                      |
| Core staple71 | CGCGAGCTCGTCTTTATTGCGA                            |
| Core staple72 | AATACTATAAGTCCTGAA                                |
| Core staple73 | TTGCACGTAATCCTCATCACGTAGTAAT                      |
| Core staple74 | GGAACCCTCGAGAGGCTTAGCGTCAATCCCCAGTCAGAAGAGAT      |
| Core staple75 | GACAAAGTAACGAAGGCACCAACCAATCGTCCATATA             |
| Core staple76 | TTGCTTTAGGAGGCTAAAGAAAAATTAT                      |
| Core staple77 | AGTTGAAAGGAGCATTTAAAAGAAACCATGGATTACATA           |
| Core staple78 | CAGCTACGCGAACAAAGTTTGAGGGTAATATAAGAATACGTGGCACA   |
| Core staple79 | GAAGGAAAGGGAGAACCAAGTTAACAAA                      |
| Core staple80 | ACAGTAGTTGGGAACTATTATCTCAAAT                      |
| Core staple81 | GCCCGAATTCTAGCATGAACGCAATAGGAACG                  |
| Core staple82 | GTGTGAAATTATGTTAGCTAA                             |
| Core staple83 | ACAGTTAGCCACCCATAGCCCTGCCGTCTCTGAAACCCTGCCGGGT    |

|                |  |
|----------------|--|
| Core staple84  | TTATAAATCAACTATATCATAGCGATTCCAAGAA                 |
| Core staple85  | TTCACAAAGCGCAGCAACTTTGAATAAGGCTTGCCATCTTGAAGTAAC   |
| Core staple86  | GGGACGACGTTTCATCAGTTGAGATTATTACACCCTCAAATCCTCAT    |
| Core staple87  | AATTGCGTAAATCAGTAGCACATGAAACCATCGATTTTCGGTTGGGTAAT |
| Core staple88  | CAACGCGCAGTATGTATTATTAGAGGAA                       |
| Core staple89  | TGAGGAAGTTTCCAACGCATATTTTAAACATTTTCG               |
| Core staple90  | CCTAGATGGGCGCATCGTACCGGTAC                         |
| Core staple91  | GCCTTAAATCAAGATTAGT                                |
| Core staple92  | GAGTAACAACGGAAGCAAAAAGGTATTCTTA                    |
| Core staple93  | GTGGCTGGCCCTCTTTCCGCGCGGCAGTCGGGCTCACACAAACGG      |
| Core staple94  | CCTGATTACTCGTAAAAACGCATTCTGGCCAACACTAAAACCTCAATAT  |
| Core staple95  | GAGGATCCACAGACGGGCAA                               |
| Core staple96  | ACATATATTCAACCGCTTCAA                              |
| Core staple97  | AGCAACAAAGTCAGAGGGTAATTGTAGGCAGTGAGAGTGGTAGCT      |
| Core staple98  | GTCCAAATCAGGACAGATGAACGACTGACCCTATTATGAGAGGGT      |
| Core staple99  | CGGGCACCGCTTCTGGTGCGGATTA                          |
| Core staple100 | GGTAAAATACGTAACAATGACACGGTGTATTTAGT                |
| Core staple101 | TTTAATTTAAAGGACCAGACGTTAGTAATTTTCATTTGGGG          |
| Core staple102 | CCTCATTTTCAGGCAACATGACCGATATGAATTT                 |
| Core staple103 | AGCATAAAGCTAGCGGATGGCTTA                           |
| Core staple104 | TCCTGAATAAACCTACTACTTCTCACCGAGGAGT                 |
| Core staple105 | CATAGCAGCACAGCAAAAAAGCCATATAAAGGAGAATT             |
| Core staple106 | GGCCTCTTGCAAGAACTCAA                               |
| Core staple107 | TGCTATTTGTATCTTTAGGAATTATGGCTA                     |
| Core staple108 | CAGAACGCTTAATGTTGAATAACAATA                        |
| Core staple109 | GATAGCCGAGATAGGCCATTGCTTTGATCAAATTATATTTTAGACA     |
| Core staple110 | TAAGCAGATAGCGATTGTAGTAATGTGTAGGTATTTAAATTATGGTT    |
| Core staple111 | ATAGAAAATTCAGCAATGCCT                              |
| Core staple112 | GAAAGACAGCATCGGACGTCACCACTCAATTCTACTATTCGTCA       |
| Core staple113 | CCTTCCTCATTGCCAGGCATTTTGAGAA                       |
| Core staple114 | GTGCGTGCATCTGCCAGTTAGGTCGAC                        |
| Core staple115 | GAGATTCGCCATTCAGGCTCGCTATTA                        |
| Core staple116 | TGTCTTTCCTTAACAGCCA                                |
| Core staple117 | GCAGATTCAAAGATAACTAAAGCATCACCTATTGATTCCCAA         |
| Core staple118 | GGGGTTTTGATGATGTACACTCATCTTTGAATCCGCCGACGACGCTGGC  |
| Core staple119 | AAATCACAATTCGAGATTGAGGGAGGGAACATAAAAACCCAAACCAATA  |
| Core staple120 | GTGTCGAACCCCAAGTGTATCACCGTCAGCTCAGTGAGGCTGGCCGTATA |
| Core staple121 | CCGCGGAATAAACAAAATAGCGTTTACCAAACAGATCAGAGCCA       |
| Core staple122 | CAAATGGTTGCTAAACAACACAGTGCCATTAGATATATGCAAGCCACCAC |
| Core staple123 | AACAGTTAATGCCCATGAAACATAAGGATAAATT                 |
| Core staple124 | TTGACCAGCTGAGATTAACACCGCCTGAGCTTGAATCGCCCTTAAACG   |
| Core staple125 | AAGAGTACTGACGAAGATGGTTGGGAA                        |
| Core staple126 | AAGAGTCGGGCGATATTAATTAACCTTGCTTCTGTTTTATTTCGGCGA   |

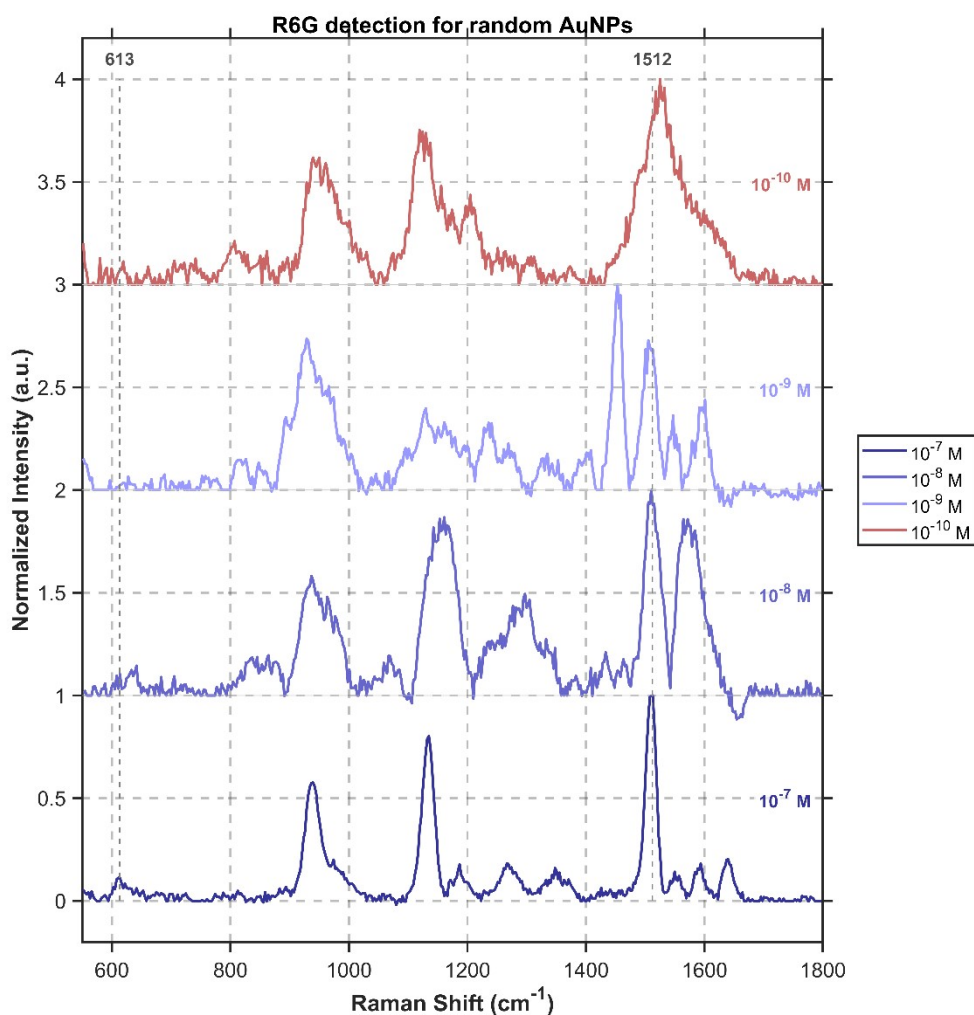
|                |  |
|----------------|--|
| Core staple127 | AGGACGTTTAATTTTCTCTGATTTGATG                       |
| Core staple128 | CAGTAGCGTAGCGCGACTGGATTTGCAA                       |
| Core staple129 | CGATTTTGAGAAAAAGATAAACACGGCCAAACCA                 |
| Core staple130 | TGCAGAAAATAGCAGCCTTAAACAACGTTATCCGAAA              |
| Core staple131 | AGCATGTAGAAATATCCCA                                |
| Core staple132 | ATATTTTGGAGAGGAAGG                                 |
| Core staple133 | GTTGTTAGAGCAGGCCAAGCGGACTGGGCGCCAGGG               |
| Core staple134 | CCAGGTAGATTTTAAAGAACTGGCGTGAATTAAGCCAGTGGTAATAA    |
| Core staple135 | TAGATAACCCTCAAATCGCCATTAATAAAAAAGGGACTCATGGAATACA  |
| Core staple136 | AGAGCAACTGTTGGGAAGGGCGATCGG                        |
| Core staple137 | ATTAATAGCGTCATAGATTGCTCGTCATC                      |
| Core staple138 | AATCGGTTGTACCAAAAACATTTTGGATAAGAGGTCATTTT          |
| Core staple139 | GTTTTGTGAAAAGGTGGCAAGACAGCCCTCATA                  |
| Core staple140 | CCATCAAAAAACAAGACCAACATGTAATTAGCGCTA               |
| Core staple141 | AATTTGCTAATATGTGTAATGGATTAGAGCGGAG                 |
| Core staple142 | TCATCAACATTGGGGTGAAGAGAAACGCTCA                    |
| Core staple143 | TGGGATTTCAATAATGTAGCTGATAGCAAGCCCAA                |
| Core staple144 | TATATTGTCACCACATTACCTATACAAAAAGTAAACAGGGAAGCG      |
| Core staple145 | GCAAGTGTAGTTTCAATTCAGATGGCGCCCAATA                 |
| Core staple146 | CGGATTGACCGTCTGTTTC                                |
| Core staple147 | AATACATAGGTAAAAATATCGCGTTTTTCATCAATAGCATGTTAAATCA  |
| Core staple148 | CAGGTCAGACGTTCCAGGGGCTTGGAAACAC                    |
| Core staple149 | TCAATTGCGTAGTAACAAGGCGAATGATGAACCTTTTTAGT          |
| Core staple150 | ACCGGAAGCACGCAAGGATA                               |
| Core staple151 | TTAGTCTGAAAGCGTCCAGAAGTGCCTTAAAAGAGCCAGAAGCTA      |
| Core staple152 | TATGCGTATTAGCACGGTTTGCATCATTGCGCTACGAGCCCCGTCGG    |
| Core staple153 | TTAATGCAAGATTAAGGTTGAATTATCATTTAA                  |
| Core staple154 | TCGCCATACCGTCACGGATTGATAAATA                       |
| Core staple155 | AACGCCAACCATCAAGCATTGACAGGATTCATAGACCTTTTCGAGAA    |
| Core staple156 | CTCGAATTGTGTCGTGCCAG                               |
| Core staple157 | ACGGCCAGATGTGGACTCCA                               |
| Core staple158 | AACTGAACACCCTGCAGTAATCCAGGCTATCAGGTGTAGCCAGCTT     |
| Core staple159 | TATTCTAAGAAACAATTTTA                               |
| Core staple160 | GGATAAGGGAATAGGCGATTAGAAAGAG                       |
| Core staple161 | GAATAATTTTCAGATTAAGGTTGGGCCAATCGCTTCTGATGATAA      |
| Core staple162 | ATAGCAATAGCTATAGAAGGAGAAAAGCAAAAGGG                |
| Core staple163 | TATGACCCTGTAATACTTTTGGAGAGTACCTTTAATTGCTCC         |
| Core staple164 | CTAAAAGCAATACTTCAACAGGTTAAATCCCGTCAA               |
| Core staple165 | TACCAGCGGAATAAAGTTACCCTTACCGAAGCCCTTTTTAAGAAAAG    |
| Core staple166 | ATTCAACAAACGAAACCACCACTTGATA                       |
| Core staple167 | TTAGAATCAGCAGGTTTGGGTTAGGAAGGCTTAT                 |
| Core staple168 | AGTGCCCTCATGGCTATTTACCGATTGGCCAGAGCTCAGCCGCCAC     |
| Core staple169 | TGGTTTTTGGAGAGAATTTTCATCAAGACAAAGAACGTAACCTCTTGAGT |

|                |   |
|----------------|---|
| Core staple170 | CGCCACCTTTCATTCTGAAATGGATTGATGCCACTCA   |
| Core staple171 | ATTCTCCGTGGGAAATTCCACCAGACGAAGTATCA   |
| Core staple172 | AGAGGGGTTTGCSCCGAATTACAGAGACTTGAATTT  |
| Core staple173 | ACGTGGTATTTTTGTACGTGAAACCAGTTTGAAC  |
| Core staple174 | AACTTTTAGCCCCCTAGGTCTGGAGGCATGATACAT  |
| Core staple175 | CAAAGGGCGAATCCTTGAATCAATCCGCACTCAT  |
| Core staple176 | CGGATTCAGACTCCAACGAGGGACCTGC  |
| Core staple177 | ATCTTTTCCGGAECTATCATAATACCAC  |
| Core staple178 | CTGTAGCATTCCCTCAGCACTCCAAAAAAGGTTTTTTCATCTAAA                                       |
| Core staple179 | ATTTTTGAGCAAAGCCGACTTTCAAGTTTG  |
| Core staple180 | ACACCACGCCAAAGGAACCCT   |
| Core staple181 | CATTAGACGGTACCGACTAAAGTGTACATT  |
| Core staple182 | AGTCACGAGCGGAACCCTAA  |
| Core staple183 | ATCACTCCAGCCAGCTTTCCAGGGTTT   |
| Core staple184 | TGATTGCCCTTTAATGGATGCTGAAATCAATAAT  |
| Core staple185 | TATTATTCCTGCAAATTTATATAAAAGAATAAAG  |
| Core staple186 | CGCTTTCGGAGAGGAGCGTTAAATAAGACCTGTTTCGACAATTACAGAG                                   |
| Core staple187 | CAGAGTGTACAATGGAACAAATAGAGCCGCGCCA  |
| Core staple188 | CCAGTACAACTACAACGC  |
| Core staple189 | TCTTCCGTAAGAAGATTATTCACGGTCACACGT   |
| Core staple190 | AACGCTACCAATATACAGATTTTAGCGGGATCCT  |
| Core staple191 | AGTTGCGGGTCAGTAGCCAGCAGCAAATCGAGTAGCTGGAAGCTCAGAAC                                  |
| Core staple192 | TCAGTGAGGCGAATAATTTTTGCGGAGGCGTTTT  |
| Core staple193 | AGCAAATATTTAAATTACAGGAACGAACAAGTTTATTTTGTACAATCA                                    |
| Core staple194 | ACAACCTTGGGAGATTTGTATCACGCGAAATATAAGTTCAGAAC  |
| Core staple195 | TTGAGGACTAAAGACTCGCTGAATAATGCCCTGTTT  |
| Core staple196 | CATATATAAGATTCSCCAAAAGTAAACG  |
| Core staple197 | AGGGAAAGAAGTAACCACGCGCAGGTACCTTTTACATCACAGAAACGATTA                                 |
| Core staple198 | AGTAAACAGTTCGGTCTTTAGCTAACTAAAGCCTAAATGTGAGC  |
| Core staple199 | ACTGAGTATAGTAGCGCTTTTAGGAGCC  |
| Core staple200 | GGTGCCGGGAAAGCTGAATTAACAAACA  |
| Core staple201 | ATCCAAACCGACCGTGCCTAAATTCACCGCTTCC  |
| Core staple202 | TCCATGTACCGTTGGCAATTCTCATATT  |
| SH_103         | HS-AAAAAAAAAAAAAAAAAAAAAAAAAAGCATAAAGCTAGCGGATGGCTTAAACAACAAC<br>AACAACAACAACAAC-SH |
| SH_150         | HS-AAAAAAAAAAAAAAAAAAAAAAAAAACCAGGACGCAAGGATAAACAACAACAACA<br>ACAACAACAACAAC-SH     |
| SH_56          | HS-AAAAAAAAAAAAATTAATATTTTGTAAATAATCAAACCGAGCGCAAAGAACAACAACA<br>C-SH               |
| SH_65          | HS-AAAAAAAAAAAAACGGCTACATAAAGGCTAGCATTCTTAATTCATGTACCGTAACAA<br>CAACAACAACAAC-SH    |

**Supplementary Table 1.** DNA sequences used in the experiments. The DONA structure was assembled from 202 staple strands and the M13mp18 scaffold strand. Four of these staples were

dual-end modified with thiol groups to replace their original counterparts for conjugating gold nanoparticles.

## 2. R6G detection for random AuNPs



**Supplementary Figure 1.** The optimal detection results for R6G concentrations ranging from 10<sup>-7</sup> M to 10<sup>-10</sup> M were obtained by depositing a 1:1 mixture of the AuNP solution and the R6G solution, followed by SERS measurement using a 638 nm laser at 1.18 mW power with a 1 s integration time. In contrast, detection using randomly aggregated AuNPs without the DNA origami scaffold showed limited performance: the detection limit was insufficient, the substrate exhibited poor reusability, and it was difficult to locate suitable scanning areas within the field of view.

## 3. Introduction to the Baseline Correction Method

Three-Step Baseline Correction and Spectral Preprocessing Pipeline

Input:

- Raw spectral data:  $\text{raw\_spectra} \in \mathbb{R}^{m \times n}$ , where  $m$  is number of spectra,  $n$  is number of wavenumber points
- Parameter structure:  $\text{params}$  (containing various thresholds and settings)

Output:

- Corrected spectra:  $\text{corrected\_spectra} \in \mathbb{R}^{m \times n}$
- Baseline estimates:  $\text{baseline1} \in \mathbb{R}^{m \times n}$ ,  $\text{baseline2} \in \mathbb{R}^{m \times n}$
- Normalized spectra:  $\text{normalized\_spectra} \in \mathbb{R}^{m \times n}$

Procedure:

1. Initialization and Preprocessing:

- 1.1 Define fingerprint region:  $\text{wavenumber\_range} \in [600, 1800] \text{ cm}^{-1}$
- 1.2 Extract fingerprint region data:  $\text{fingerprint\_data} \leftarrow \text{raw\_spectra}(\text{fingerprint\_region})$
- 1.3 Optional: Apply Savitzky-Golay filter for data smoothing

2. Three-Step Baseline Correction (executed for each spectrum):

for  $i = 1$  to  $m$  do  
     $\text{spectrum} \leftarrow \text{fingerprint\_data}(i, :)$

    Step A: Morphological Baseline Correction

        Input:  $\text{signal} \leftarrow \text{spectrum}$ ,  $\text{params}$

        Process:

- a. Set structuring element length:  $L \leftarrow$  if  $\text{params.use\_fixed\_length}$  then 81 else 51
  - b. Initialize:  $\text{baseline\_prev} \leftarrow \text{signal}$ ,  $\text{baseline\_current} \leftarrow \text{signal}$
  - c. for iteration = 1 to  $\text{params.max\_iterations}$  do
    - i. Create linear structuring element:  $\text{SE} \leftarrow \text{strel}(\text{'line'}, L, 0)$
    - ii. Morphological opening:  $\text{opening\_op} \leftarrow \text{imopen}(\text{baseline\_prev}, \text{SE})$
    - iii. Morphological closing:  $\text{closing\_op} \leftarrow \text{imclose}(\text{baseline\_prev}, \text{SE})$
    - iv. Compute average:  $\text{avg\_op} \leftarrow (\text{opening\_op} + \text{closing\_op})/2$
    - v. Update baseline:  $\text{baseline\_current} \leftarrow \min(\text{signal}, \text{avg\_op})$
    - vi. Smooth baseline:  $\text{baseline\_current} \leftarrow \text{gaussian\_smooth}(\text{baseline\_current}, \text{window\_size}=L/2)$
  - vii. Check convergence:  
        if  $\text{iteration} > 1$  and  $\frac{\sum(\text{baseline\_current} - \text{baseline\_prev})^2}{\sum(\text{baseline\_prev})^2} < \text{params.convergence\_threshold}$   
        then break loop
  - viii.  $\text{baseline\_prev} \leftarrow \text{baseline\_current}$
- d.  $\text{morphological\_baseline} \leftarrow \text{baseline\_current}$
- e.  $\text{preliminary\_corrected\_signal} \leftarrow \text{signal} - \text{morphological\_baseline}$

    Step B: Optimized Piecewise Baseline Correction

Input: signal  $\leftarrow$  preliminary\_corrected\_signal

Process:

- a. Smooth signal: smoothed\_signal  $\leftarrow$  moving\_average(signal, window\_size=5)
- b. Find local minima: [minima, min\_locations]  $\leftarrow$  findpeaks(-smoothed\_signal)
- c. Ensure boundary points included:
  - if min\_locations(1)  $\neq$  1 then add smoothed\_signal(1) at position 1
  - if min\_locations(end)  $\neq$  n then add smoothed\_signal(n) at position n
- d. Linear interpolation: piecewise\_baseline  $\leftarrow$  interp1(min\_locations, minima, 1:n, 'linear')
- e. Smooth piecewise baseline: piecewise\_baseline  $\leftarrow$  moving\_average(piecewise\_baseline, window\_size=11)
- f. Ensure baseline doesn't exceed signal: piecewise\_baseline  $\leftarrow$  min(piecewise\_baseline, smoothed\_signal)
- g. secondary\_corrected\_signal  $\leftarrow$  signal - piecewise\_baseline
- h. Ensure non-negative: secondary\_corrected\_signal  $\leftarrow$  secondary\_corrected\_signal - min(secondary\_corrected\_signal)

Step C: Normalization and Thresholding

Process:

- a. Max-value normalization:
  - if max(secondary\_corrected\_signal) > 0
  - then normalized\_signal  $\leftarrow$  secondary\_corrected\_signal / max(secondary\_corrected\_signal)
  - else normalized\_signal  $\leftarrow$  secondary\_corrected\_signal
- b. Intensity thresholding:
  - threshold  $\leftarrow$  params.normalized\_intensity\_threshold \* max(normalized\_signal)
  - normalized\_signal(normalized\_signal < threshold)  $\leftarrow$  0
- c. Forced zeroing:
  - low\_threshold  $\leftarrow$  threshold  $\times$  0.5
  - normalized\_signal(normalized\_signal < low\_threshold)  $\leftarrow$  0
- d. Optional: Apply Savitzky-Golay filter to normalized signal

Save Results:

```
corrected_spectra(:, i)  $\leftarrow$  normalized_signal
baseline1(:, i)  $\leftarrow$  morphological_baseline
baseline2(:, i)  $\leftarrow$  piecewise_baseline
end for
```

3. Calculate Average Spectrum:

```
average_spectrum  $\leftarrow$  mean(corrected_spectra, dimension=2)
Repeat Step C normalization and thresholding for average_spectrum
```

4. Output:

```
Return corrected_spectra, baseline1, baseline2, normalized_spectra
```