Supporting Information

Spectroscopic and kinetic characterization of the diheme cytochrome c peroxidase from *Shewanella oneidensis*

Gökçe Su Pulcu, Katherine E. Frato, Rupal Gupta, Hao-Ru Hsu, George A. Levine, Michael P. Hendrich and Sean J. Elliott[*]

Materials and Methods.

Mutagenesis of So CcP. Mutagenesis was performed using the QuikChange protocol according the manufacturer's instructions. Mutations are shown in bold, and reverse primers were the exact reverse and complement of the primers given. The P75T/H81K/E84Q variant was made by first adding the P75T mutation using the primer CGGCGTAGATGCGCTA**ACC**ACGTCAATTGGT. Then, the H81K/E84Q double mutation was added simultaneously using the primer ACGTCAATTGGTCAT**AAG**TGGCAA**CAA**GGC. The M219Q/F247N variant was also constructed by step-wise mutation. First, the M219Q mutation was added using the primer CCCGCGGTAGGCGGTACT**CAG**TTTATGAAGATGGGCCTAATTAAAC, followed by addition of the F247N mutation using the primer CGGGTAAGGATGCGGATAAG**AAC**GTGTTTAAGGTACCGACACTGC.

Lag-phase least-square fitting program.

```
# optimize_lagphase.py

# Initialize a dictionary to hold the experimental data

data = {}
print data

f = open (path/datafile)

for line in f:
        data_string = str(line)
        print data_string
        print len(data_string)
        if len(data_string)>5 and data_string.split('.')[0].isdigit():
                t = data_string.split(',')[0]
                round_time = round(float(t), 1)
                print t, round_time
                absorbance = data_string.split(',')[1]
                print absorbance
                data[str(round_time)]=absorbance
f.close()

print data

# This function defines the residuals, ie the data minus the model

def residuals(p, data_dict):
        t = 0.1
        CcP_ox = 0.016              #Initialize to the initial peroxidase concentration
        CytC_initial = 40.40        #Initialize to the initial electron donor concentration
        CcP_act = 0                 #All the peroxidase begins in the oxidized, inactive state
        CytC_ox = 0                 #All the electron donor begins in the reduced state
        delta_CcP_act = 0
        delta_CytC = 0
        CcP_total = float(CcP_ox)
        # CytC_red_stuck = p[2]
        CcP_monomer = 0             #All the peroxidase begins in a dimeric state

        # Input rates per second and Kd in uM. **Change these to make the model fit.**

        Kd = 7
        kact_s = p[1]
        k1_s = p[0]
        k_dissoc_s = p[2]           #dissociation of the dimer over time
        curve = []
```

```
        # Rates must be per 0.2 seconds to match the time step of integration
        kact = float(kact_s)/5.0
        k1 = float(k1_s)/5.0
        k_dissoc = float(k_dissoc_s)/5.0
        CytC_red = float(CytC_initial)

        for number in range(2200):
                # Calculate the fraction of CcP (total) bound to CytC5
                fraction_bound = float(CytC_red)/(float(CytC_red) + float(Kd))
                #print 'fraction bound', fraction_bound

                # There is some probability that, once bound, the CcP will activate
                delta_CcP_act = float(kact)*float(fraction_bound)*float(CcP_ox)
                #print 'change in CcP activation', delta_CcP_act
                CcP_monomer += float(CcP_act)*k_dissoc
                CcP_ox -= float(delta_CcP_act)
                CcP_act += float(delta_CcP_act)-float(CcP_act)*k_dissoc
                print 'total inactive and active CcP', CcP_ox, CcP_act, CcP_monomer
                #print CcP_monomer

                # Calculate the amount of CytC(red) used up in this time step
                delta_CytC = float(delta_CcP_act)+ \
                 2*float(k1)*float(CcP_act)*float(CytC_red)
                CytC_ox += float(delta_CytC)
                CytC_red -=float(delta_CytC)
                delta_CcP_act = 0

                # Calculate the absorbance at 553
                A = float(CytC_ox)*0.0084 + float(CytC_red)*0.0295
                t_ref = str(round(t, 1))
                curve.append([t_ref,A])
                t += 0.2

        # Calculate difference between absorbance in data file and generated by the model
        errors = []
        for pair in curve:
                try:
                        data_value = data_dict[pair[0]]
                        errors.append(float(pair[1])-float(data_value))
                except KeyError:
                        #print "Got to the end of the data set"
                        continue
        return errors

# Give an initial set of guesses [k1, kact, k_dissoc]
```

```
p0 = [0.1, 0.01, 0.001]
import numpy

from scipy.optimize import leastsq
plsq = leastsq(residuals, p0[:], args=(data))
print plsq[0]
```

Results

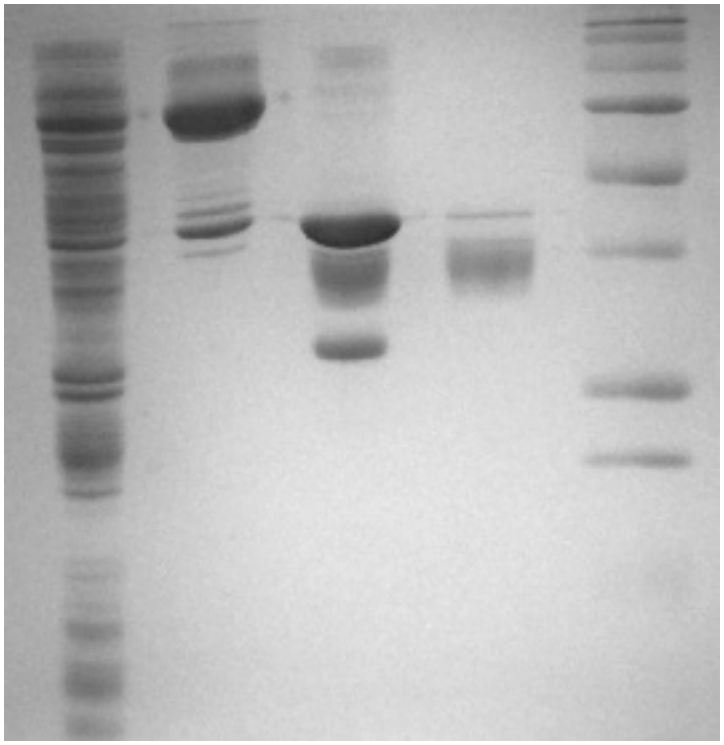

Figure S1: Coomassie-stained 16% SDS-PAGE gel of the purification of *So* C*c*P. From left to right, Lane 1: soluble fraction of the lysate; Lane 2: Flow-through from amylose resin; Lane 3: Flow-through from the amylose column, digested overnight with TEV protease; Lane 4: Sample following Ni-NTA chromatography to remove MBP tag and TEV.

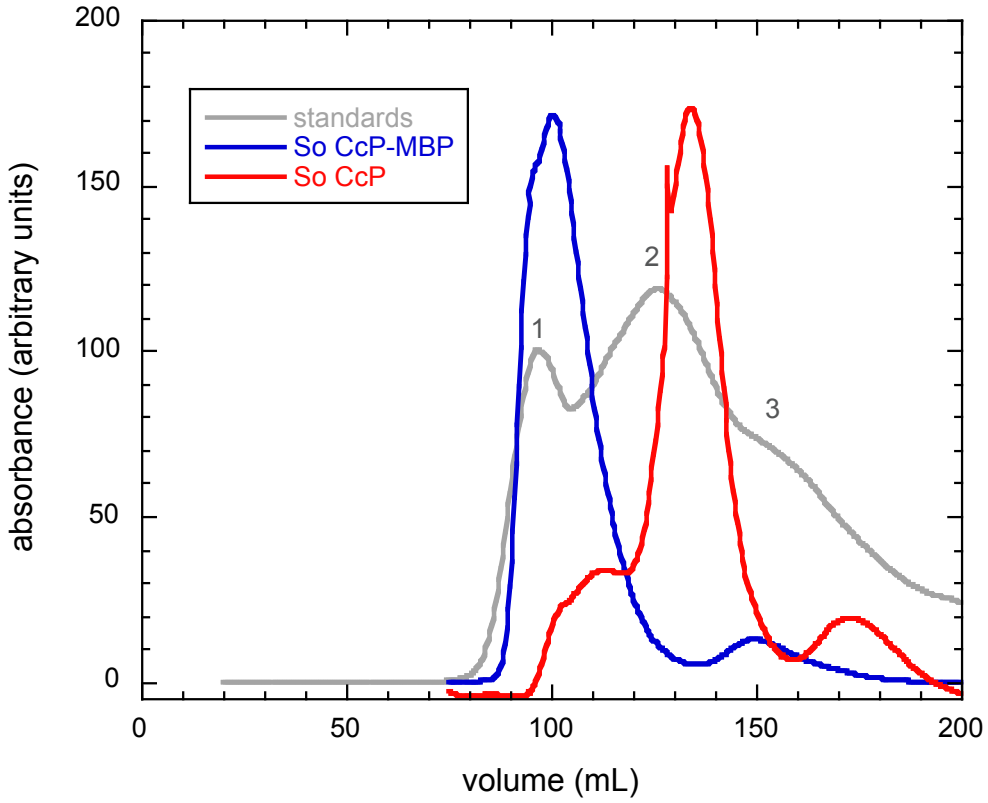

Figure S2: Size exclusion chromatography of *So* C*c*P-MBP (blue) compared to tag-free *So* C*c*P (red). Grey trace shows molecular weight standards; 1: blue dextran (exclusion volume), 2: albumin (66 kDa), 3: carbonic anhydrase (29 kDa).

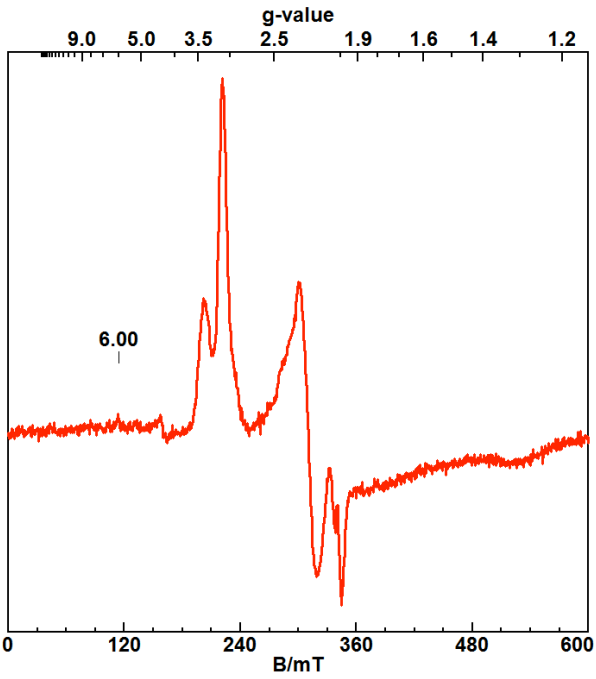

Figure S3: EPR spectra of 0.5 mM MBP-C*c*P in the absence of L-ascorbate. Experiments were conduced with microwave frequency of 9.62 GHz, power of 0.02 mW, and temperature of 7 K.

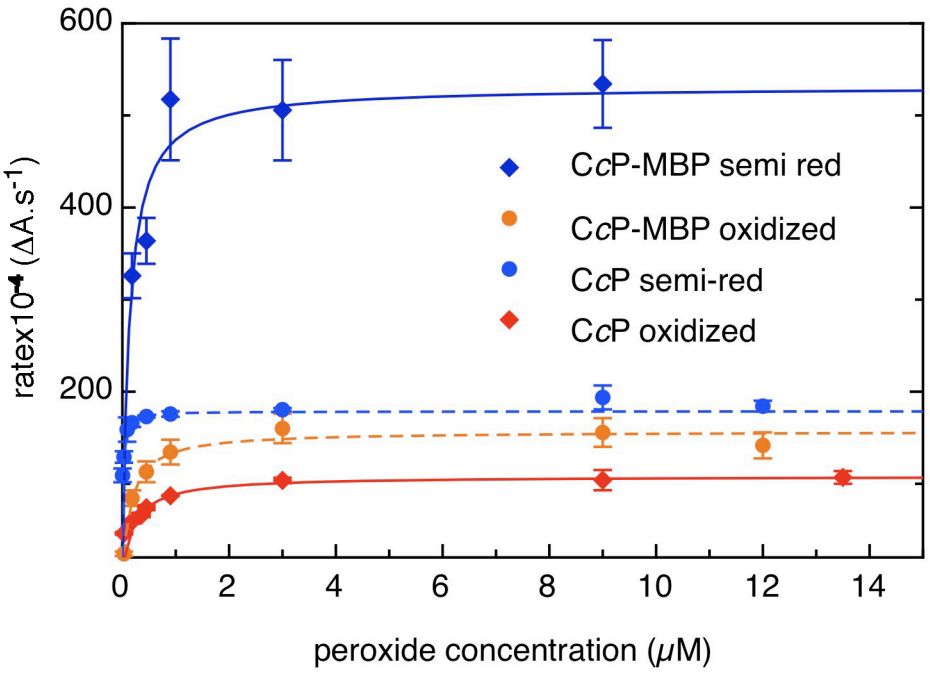

Figure S4: Michaelis-Menten analysis of peroxide turnover of *So* C*c*P-MBP in the semi-reduced (blue, solid) or oxidized (orange, dashed) state, and tag-free *So* C*c*P in the semi-reduced (blue, dashed) or oxidized (red, solid) state. Assays were conducted with 10 µM horse heart cytochrome c and 5 nM enzyme in buffer containing 5 mM HEPES, 5 mM MES, 10 mM NaCl and 1 mM $CaCl_2$ at 23°C.

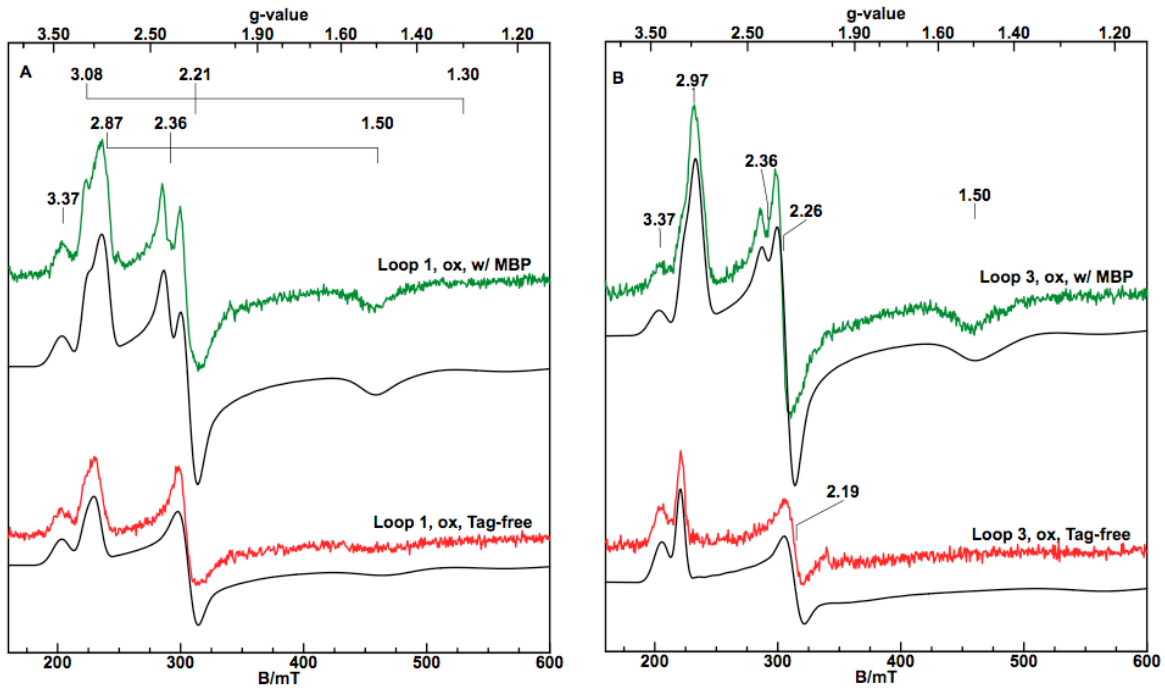

Figure S5: (A) EPR characterization of Loop 1 mutants as the oxidized, MBP-fused protein (green) or tag-free/fully oxidized protein (red). Simulations based upon the contributions from states tabulated in Table S1 are shown in black lines. Data were collected as in Figure 4. (B) The analogous EPR spectra for the Loop 3 mutants.

Table S1: EPR contributions of the Loop 1 and Loop 3 mutants.

Heme Species	Amount with respect to total Fe (%)			
	MBP-Loop1 mutant, ox	Loop1 mut, tag-free	MBP-Loop3 mutant, ox	Loop3 mut, tag-free
H-Heme g= (3.37, 2.20, 1.50)	21	29	15	45
L_1-Heme g = (3.08 ,2.26, 1.30)	29	36	27	55 g = (3.12,2.19 1.30)
L_2-Heme g = (2.87, 2.36, 1.51)	17	0	10	0
TF-Heme g = (2.97, 2.24, 1.52)	32	34	48	0