Supporting information for
Restriction-based Multiple-fragment Assembly Strategy for Long Coding Sequence Cloning from complementary DNA

Shang Wang[1,2#], Wen Chen[1,2#], Kai Zhang[1,2#], Peng Jiao[1,2], Lihua Mo[1,2], Xiaoxu Yang[1,2], Xiang Hu[1,2], Jian Zhang[1,2], Chenxi Wei[1,2*], Shuanglin Xiang[1,2*]

[1.] Key Laboratory of Protein Chemistry and Developmental Biology of Education Ministry of China, College of Life Sciences, Hunan Normal University, Changsha 410081, China;

[2.] The Cooperative Innovation Center of Engineering and New Products for Developmental Biology of Hunan Province (20134486), Changsha 410081, China

[#] These authors contributed equally to this work.

[*]Corresponding author. Tel.: +86 731 88872095/88872916. Fax: 86-731-8887-2905

E-mail address: weicx@hunnu.edu.cn; xshlin@hunnu.edu.cn

The authors have declared that no competing interests exist.

Short title: Multiple-fragment Assembly Strategy for long CDS Cloning

Category: DNA Recombinant Techniques and Nucleic Acids

S1. Python scripts

Human.py

```
# -*- coding: utf-8 -*-
import os
CDSs = open("Human_CCDS_nucleotide.current.fna", "r")
Results = open("Human_Results.txt","w")
CDS_OK = open("Human_CDS_OK.txt", "w")
CDS_Name =[]
CDS_Seq =[]
CDS_Num = -1
for line in CDSs.readlines():
    line=line.rstrip()
    if '>' in line:
        CDS_Name.append(line)
        CDS_Num = CDS_Num + 1
        CDS_Seq.append("")
    else:
        CDS_Seq[CDS_Num] = CDS_Seq[CDS_Num] + line

CDS_more1500 = 0
CDS_less1500 = 0
Can_not_cut = 0
Can_cut = 0
enzyme = {"ApaI":"GGGCCC", "BamHI":"GGATCC", "BglII":"AGATCT", "EcoRI":"GAATTC",
"HindIII":"AAGCTT", "KpnI":"GGTACC", "NcoI":"CCATGG", "NdeI":"CATATG",
"NheI":"GCTAGC", "NotI":"GCGGCCGC", "SacI":"GAGCTC", "SalI":"GTCGAC", "SphI":"GCATGC",
"XbaI":"TCTAGA", "XhoI":"CTCGAG"}
for line in CDS_Seq:
    if len(line) >= 1500:
        CDS_more1500 = CDS_more1500 + 1
        enzymesite = [len(line),0]
        for val in enzyme.values():
            if line.count(val) == 1:
                enzymesite.append(line.find(val))
        enzymesite.sort(reverse = True)
        Bad = False
        for i in range(len(enzymesite)-1):
            j = enzymesite[i] - enzymesite[i+1]
            if j > 1500:
                Bad = True
        if Bad:
            Can_not_cut = Can_not_cut + 1
        else:
            Can_cut = Can_cut + 1
            CDS_OK.write(CDS_Name[CDS_Seq.index(line)] + "\n")
            CDS_OK.write(line + "/n")
    else:
        CDS_less1500 = CDS_less1500 + 1

print "Total CDS: " + str(len(CDS_Name))
print "CDS less than 1500: " + str(CDS_less1500)
print "CDS more than 1500: " + str(CDS_more1500)
print "Can cut: " + str(Can_cut)
print "Can not cut: " +str(Can_not_cut)

Results.write("Total CDS: " + str(len(CDS_Name)) + "\n")
Results.write("CDS less than 1500: " + str(CDS_less1500) + "\n")
Results.write("CDS more than 1500: " + str(CDS_more1500) + "\n")
Results.write("Can cut: " + str(Can_cut) + "\n")
Results.write("Can not cut: " +str(Can_not_cut) + "\n")
```

Mouse.py

```
# -*- coding: utf-8 -*-
import os
CDSs = open("Mouse_CCDS_nucleotide.current.fna", "r")
Results = open("Mouse_Results.txt","w")
CDS_OK = open("Mouse_CDS_OK.txt", "w")
CDS_Name =[]
CDS_Seq =[]
CDS_Num = -1
for line in CDSs.readlines():
    line=line.rstrip()
    if '>' in line:
        CDS_Name.append(line)
        CDS_Num = CDS_Num + 1
        CDS_Seq.append("")
    else:
        CDS_Seq[CDS_Num] = CDS_Seq[CDS_Num] + line

CDS_more1500 = 0
CDS_less1500 = 0
Can_not_cut = 0
Can_cut = 0
enzyme = {"ApaI":"GGGCCC", "BamHI":"GGATCC", "BglII":"AGATCT", "EcoRI":"GAATTC",
"HindIII":"AAGCTT", "KpnI":"GGTACC", "NcoI":"CCATGG", "NdeI":"CATATG",
"NheI":"GCTAGC", "NotI":"GCGGCCGC", "SacI":"GAGCTC", "SalI":"GTCGAC", "SphI":"GCATGC",
"XbaI":"TCTAGA", "XhoI":"CTCGAG"}
for line in CDS_Seq:
    if len(line) >= 1500:
        CDS_more1500 = CDS_more1500 + 1
        enzymesite = [len(line),0]
        for val in enzyme.values():
            if line.count(val) == 1:
                enzymesite.append(line.find(val))
        enzymesite.sort(reverse = True)
        Bad = False
        for i in range(len(enzymesite)-1):
            j = enzymesite[i] - enzymesite[i+1]
            if j > 1500:
                Bad = True
        if Bad:
            Can_not_cut = Can_not_cut + 1
        else:
            Can_cut = Can_cut + 1
            CDS_OK.write(CDS_Name[CDS_Seq.index(line)] + "\n")
            CDS_OK.write(line + "/n")
    else:
        CDS_less1500 = CDS_less1500 + 1

print "Total CDS: " + str(len(CDS_Name))
print "CDS less than 1500: " + str(CDS_less1500)
print "CDS more than 1500: " + str(CDS_more1500)
print "Can cut: " + str(Can_cut)
print "Can not cut: " +str(Can_not_cut)

Results.write("Total CDS: " + str(len(CDS_Name)) + "\n")
Results.write("CDS less than 1500: " + str(CDS_less1500) + "\n")
Results.write("CDS more than 1500: " + str(CDS_more1500) + "\n")
Results.write("Can cut: " + str(Can_cut) + "\n")
Results.write("Can not cut: " +str(Can_not_cut) + "\n")
```

Human_len.py

```
# -*- coding: utf-8 -*-
import os
CDSs = open("Human_CCDS_nucleotide.current.fna", "r")
CDS_len = open("Human_len.txt","w")
CDS_Name =[]
CDS_Seq =[]
CDS_Num = -1
for line in CDSs.readlines():
    line=line.rstrip()
    if '>' in line:
        CDS_Name.append(line)
        CDS_Num = CDS_Num + 1
        CDS_Seq.append("")
    else:
        CDS_Seq[CDS_Num] = CDS_Seq[CDS_Num] + line

CDS_less500 = 0
CDS_500_1000 = 0
CDS_1000_1500 = 0
CDS_more1500 = 0
for line in CDS_Seq:
    if len(line) < 500:
        CDS_less500 = CDS_less500 + 1
    elif len(line) < 1000:
        CDS_500_1000 = CDS_500_1000 +1
    elif len(line) < 1500:
        CDS_1000_1500 = CDS_1000_1500 + 1
    else:
        CDS_more1500 = CDS_more1500 + 1

print "CDS less than 500: " + str(CDS_less500)
print "CDS 500 - 1000: " + str(CDS_500_1000)
print "CDS 1000 -1500: " + str(CDS_1000_1500)
print "CDS more than 1500: " + str(CDS_more1500)

CDS_len.write("CDS less than 500: " + str(CDS_less500) + "\n")
CDS_len.write("CDS 500 - 1000: " + str(CDS_500_1000) + "\n")
CDS_len.write("CDS 1000 -1500: " + str(CDS_1000_1500) + "\n")
CDS_len.write("CDS more than 1500: " + str(CDS_more1500) + "\n")
```

Mouse_len.py

```
# -*- coding: utf-8 -*-
import os
CDSs = open("Mouse_CCDS_nucleotide.current.fna", "r")
Results = open("Mouse_Results.txt","w")
CDS_OK = open("Mouse_CDS_OK.txt", "w")
CDS_Name =[]
CDS_Seq =[]
CDS_Num = -1
for line in CDSs.readlines():
    line=line.rstrip()
    if '>' in line:
        CDS_Name.append(line)
        CDS_Num = CDS_Num + 1
        CDS_Seq.append("")
    else:
        CDS_Seq[CDS_Num] = CDS_Seq[CDS_Num] + line

CDS_more1500 = 0
CDS_less1500 = 0
Can_not_cut = 0
Can_cut = 0
enzyme = {"ApaI":"GGGCCC", "BamHI":"GGATCC", "BglII":"AGATCT", "EcoRI":"GAATTC",
"HindIII":"AAGCTT", "KpnI":"GGTACC", "NcoI":"CCATGG", "NdeI":"CATATG",
"NheI":"GCTAGC", "NotI":"GCGGCCGC", "SacI":"GAGCTC", "SalI":"GTCGAC", "SphI":"GCATGC",
"XbaI":"TCTAGA", "XhoI":"CTCGAG"}
for line in CDS_Seq:
    if len(line) >= 1500:
        CDS_more1500 = CDS_more1500 + 1
        enzymesite = [len(line),0]
        for val in enzyme.values():
            if line.count(val) == 1:
                enzymesite.append(line.find(val))
        enzymesite.sort(reverse = True)
        Bad = False
        for i in range(len(enzymesite)-1):
            j = enzymesite[i] - enzymesite[i+1]
            if j > 1500:
                Bad = True
        if Bad:
            Can_not_cut = Can_not_cut + 1
        else:
            Can_cut = Can_cut + 1
            CDS_OK.write(CDS_Name[CDS_Seq.index(line)] + "\n")
            CDS_OK.write(line + "/n")
    else:
        CDS_less1500 = CDS_less1500 + 1

print "Total CDS: " + str(len(CDS_Name))
print "CDS less than 1500: " + str(CDS_less1500)
print "CDS more than 1500: " + str(CDS_more1500)
print "Can cut: " + str(Can_cut)
print "Can not cut: " +str(Can_not_cut)

Results.write("Total CDS: " + str(len(CDS_Name)) + "\n")
Results.write("CDS less than 1500: " + str(CDS_less1500) + "\n")
Results.write("CDS more than 1500: " + str(CDS_more1500) + "\n")
Results.write("Can cut: " + str(Can_cut) + "\n")
Results.write("Can not cut: " +str(Can_not_cut) + "\n")
```

S2

Length distribution of Human and Mouse CDS

	Human	**mouse**
0~500bp	3015(10.37%)	2287(9.58%)
500~1000bp	7288(25.08%)	6835(28.63%)
1000~1500bp	6643(22.86%)	5273(22.09%)
1500bp~max	12118(41.69%)	9479(39.70%)
total	29064	23874

Human and mouse CDSs were analyzed in Human_len.py and Mouse_len.py.

S3

CDS fits Restriction-based Multiple-fragment Assembly Strategy

	Human	**mouse**
output	8304(68.52%)	6678(70.45%)
total	12118	9479

CDS records longer than 1,500bp are analyzed in Human.py and Mouse.py.