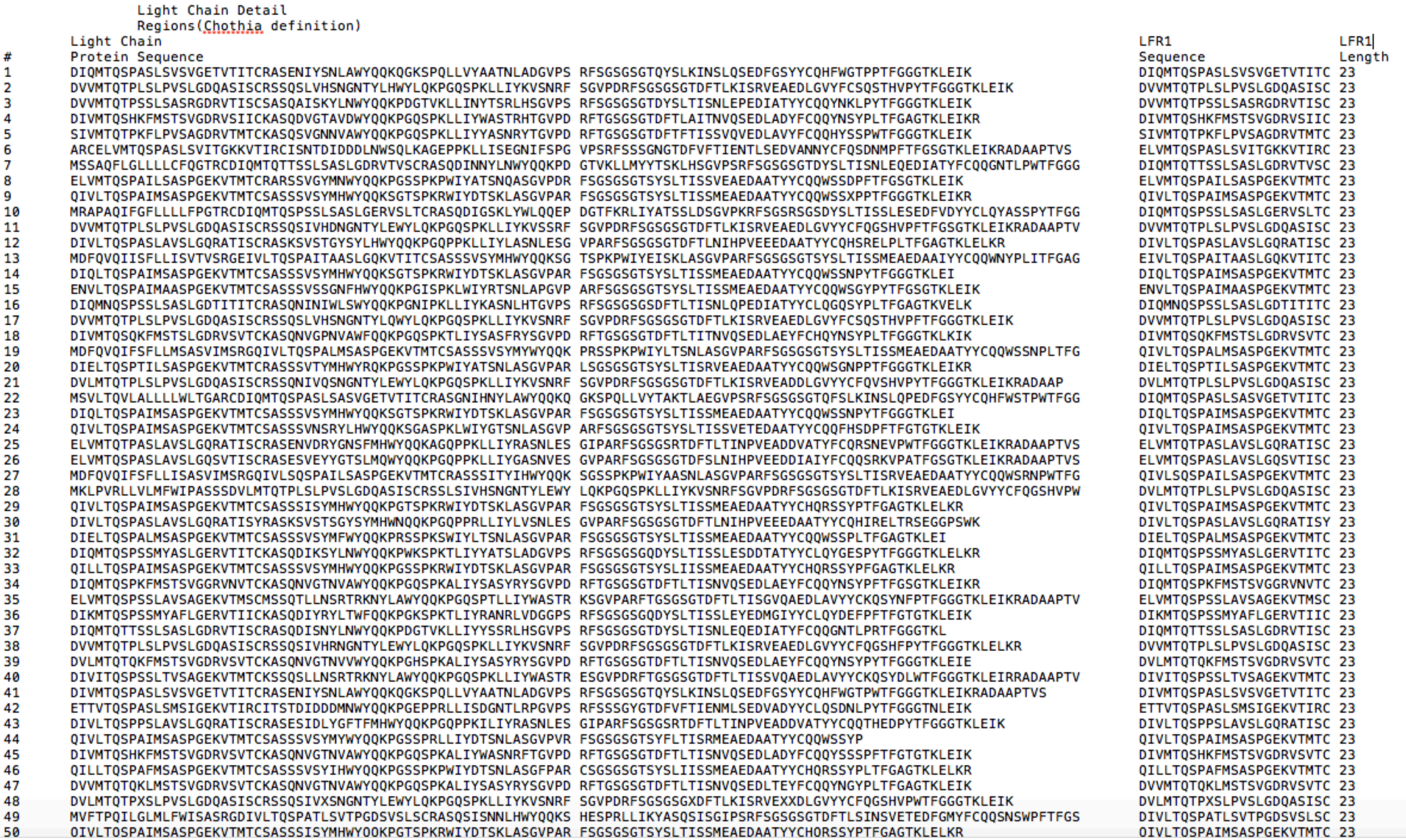
Supplementary Material

Supplemental Figure 1. AbYsis database search and analysis.

Page 1 shows the first 50 sequences (of 2676) are shown, including the full length sequence followed by the FR1 sequence (defined using Chothia numbering system).

Pages 2 and 3 show the python script used to process and analyze FR1 sequences and the output from running the script.

Pages 4 and 5 show the python script used to process and analyze SP sequences and the output from running the script.



Light Chain Detail

Regions(Chothia definition)

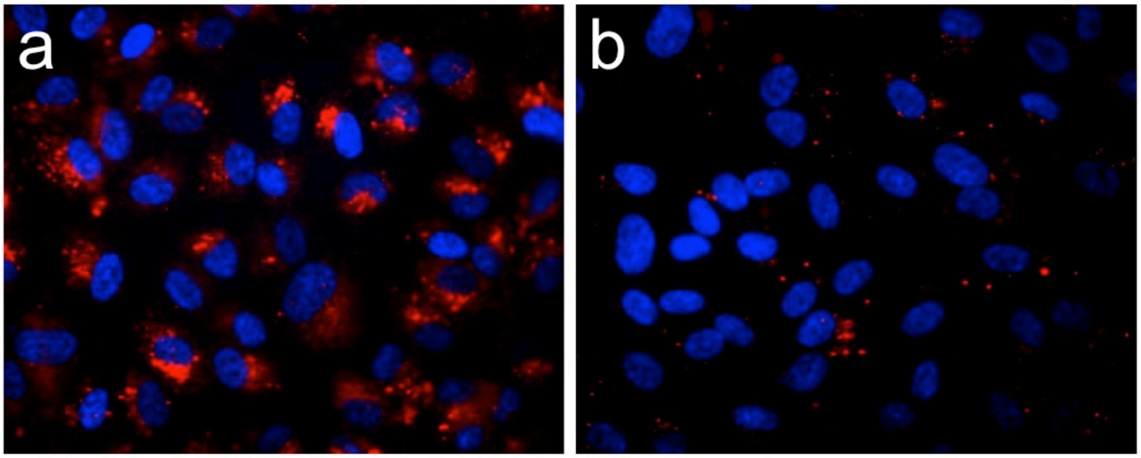
#	Light Chain Protein Sequence	LFR1 Sequence	LFR1 Length
1	DIQMTQSPASLSVSVGETVTITCRASENIYSNLAWYQQKQGKSPQLLVYAATNLADGVPS RFSGSGSGTQYSLKINSLQSEDFGSYYCQHFWGTPPTFGGGTKLEIK	DIQMTQSPASLSVSVGETVTITC	23
2	DVVMTQTPLSLPVSLGDQASISCRSSQSLVHSNGNTYLHWYLQKPGQSPKLLIYKVSNRF SGVPDRFSGSGSGTDFTLKISRVEAEDLGVYFCSQSTHVPYTFGGGTKLEIK	DVVMTQTPLSLPVSLGDQASISC	23
3	DVVMTQTPSSLSASRGDRVTISCSASQAISKYLNWYQQKPDGTVKLLINYTSRLHSGVPS RFSGSGSGTDYSLTISNLEPEDIATYYCQQYNKLPYTFGGGTKLEIK	DVVMTQTPSSLSASRGDRVTISC	23
4	DIVMTQSHKFMSTSVGDRVSIICKASQDVGTAVDWYQQKPGQSPKLLIYWASTRHTGVPD RFTGSGSGTDFTLAITNVQSEDLADYFCQQYNSYPLTFGAGTKLEIKR	DIVMTQSHKFMSTSVGDRVSIIC	23
5	SIVMTQTPKFLPVSAGDRVTMTCKASQSVGNNVAWYQQKPGQSPKLLIYYASNRYTGVPD RFTGSGSGTDFTFTISSVQVEDLAVYFCQQHYSSPWTFGGGTKLEIK	SIVMTQTPKFLPVSAGDRVTMTC	23
6	ARCELVMTQSPASLSVITGKKVTIRCISNTDIDDDLNWSQLKAGEPPKLLISEGNIFSPG VPSRFSSSGNGTDFVFTIENTLSEDVANNYCFQSDNMPFTFGSGTKLEIKRADAAPTVS	ELVMTQSPASLSVITGKKVTIRC	23
7	MSSAQFLGLLLLCFQGTRCDIQMTQTTSSLSASLGDRVTVSCRASQDINNYLNWYQQKPD GTVKLLMYYTSKLHSGVPSRFSGSGSGTDYSLTISNLEQEDIATYFCQQGNTLPWTFGGG	DIQMTQTTSSLSASLGDRVTVSC	23
8	ELVMTQSPAILSASPGEKVTMTCRARSSVGYMNWYQQKPGSSPKPWIYATSNQASGVPDR FSGSGSGTSYSLTISSVEAEDAATYYCQQWSSDPFTFGSGTKLEIK	ELVMTQSPAILSASPGEKVTMTC	23
9	QIVLTQSPAIMSASPGEKVTMTCSASSSVSYMHWYQQKSGTSPKRWIYDTSKLASGVPAR FSGSGSGTSYSLTISSMEAEDAATYYCQQWSSXPPTFGGGTKLEIKR	QIVLTQSPAIMSASPGEKVTMTC	23
10	MRAPAQIFGFLLLLFPGTRCDIQMTQSPSSLSASLGERVSLTCRASQDIGSKLYWLQQEP DGTFKRLIYATSSLDSGVPKRFSGSRSGSDYSLTISSLESEDFVDYYCLQYASSPYTFGG	DIQMTQSPSSLSASLGERVSLTC	23
11	DVVMTQTPLSLPVSLGDQASISCRSSQSIVHDNGNTYLEWYLQKPGQSPKLLIYKVSSRF SGVPDRFSGSGSGTDFTLKISRVEAEDLGVYYCFQGSHVPFTFGSGTKLEIKRADAAPTV	DVVMTQTPLSLPVSLGDQASISC	23
12	DIVLTQSPASLAVSLGQRATISCRASKSVSTGYSYLHWYQQKPGQPPKLLIYLASNLESG VPARFSGSGSGTDFTLNIHPVEEEDAATYYCQHSRELPLTFGAGTKLELKR	DIVLTQSPASLAVSLGQRATISC	23
13	MDFQVQIISFLLISVTVSRGEIVLTQSPAITAASLGQKVTITCSASSSVSYMHWYQQKSG TSPKPWIYEISKLASGVPARFSGSGSGTSYSLTISSMEAEDAAIYYCQQWNYPLITFGAG	EIVLTQSPAITAASLGQKVTITC	23
14	DIQLTQSPAIMSASPGEKVTMTCSASSSVSYMHWYQQKSGTSPKRWIYDTSKLASGVPAR FSGSGSGTSYSLTISSMEAEDAATYYCQQWSSNPYTFGGGTKLEI	DIQLTQSPAIMSASPGEKVTMTC	23
15	ENVLTQSPAIMAASPGEKVTMTCSASSSVSSGNFHWYQQKPGISPKLWIYRTSNLAPGVP ARFSGSGSGTSYSLTISSMEAEDAATYYCQQWSGYPYTFGSGTKLEIK	ENVLTQSPAIMAASPGEKVTMTC	23
16	DIQMNQSPSSLSASLGDTITITCRASQNINIWLSWYQQKPGNIPKLLIYKASNLHTGVPS RFSGSGSGSDFTLTISNLQPEDIATYYCLQGQSYPLTFGAGTKVELK	DIQMNQSPSSLSASLGDTITITC	23
17	DVVMTQTPLSLPVSLGDQASISCRSSQSLVHSNGNTYLQWYLQKPGQSPKLLIYKVSNRF SGVPDRFSGSGSGTDFTLKISRVEAEDLGVYFCSQSTHVPFTFGGGTKLEIK	DVVMTQTPLSLPVSLGDQASISC	23
18	DIVMTQSQKFMSTSLGDRVSVTCKASQNVGPNVAWFQQKPGQSPKTLIYSASFRYSGVPD RFTGSGSGTDFTLTITNVQSEDLAEYFCHQYNSYPLTFGGGTKLKIK	DIVMTQSQKFMSTSLGDRVSVTC	23
19	MDFQVQIFSFLLMSASVIMSRGQIVLTQSPALMSASPGEKVTMTCSASSSVSYMYWYQQK PRSSPKPWIYLTSNLASGVPARFSGSGSGTSYSLTISSMEAEDAATYYCQQWSSNPLTFG	QIVLTQSPALMSASPGEKVTMTC	23
20	DIELTQSPTILSASPGEKVTMTCRASSSVTYMHWYRQKPGSSPKPWIYATSNLASGVPAR LSGSGSGTSYSLTISRVEAEDAATYYCQQWSGNPPTFGGGTKLEIKR	DIELTQSPTILSASPGEKVTMTC	23
21	DVLMTQTPLSLPVSLGDQASISCRSSQNIVQSNGNTYLEWYLQKPGQSPKLLIYKVSNRF SGVPDRFSGSGSGTDFTLKISRVEADDLGVYYCFQVSHVPYTFGGGTKLEIKRADAAP	DVLMTQTPLSLPVSLGDQASISC	23
22	MSVLTQVLALLLLWLTGARCDIQMTQSPASLSASVGETVTITCRASGNIHNYLAWYQQKQ GKSPQLLVYTAKTLAEGVPSRFSGSGSGTQFSLKINSLQPEDFGSYYCQHFWSTPWTFGG	DIQMTQSPASLSASVGETVTITC	23
23	DIQLTQSPAIMSASPGEKVTMTCSASSSVSYMHWYQQKSGTSPKRWIYDTSKLASGVPAR FSGSGSGTSYSLTISSMEAEDAATYYCQQWSSNPYTFGGGTKLEI	DIQLTQSPAIMSASPGEKVTMTC	23
24	QIVLTQSPAIMSASPGEKVTMTCSASSSVNSRYLHWYQQKSGASPKLWIYGTSNLASGVP ARFSGSGSGTSYSLTISSVETEDAATYYCQQFHSDPFTFGTGTKLEIK	QIVLTQSPAIMSASPGEKVTMTC	23
25	ELVMTQTPASLAVSLGQRATISCRASENVDRYGNSFMHWYQQKAGQPPKLLIYRASNLES GIPARFSGSGSRTDFTLTINPVEADDVATYFCQRSNEVPWTFGGGTKLEIKRADAAPTVS	ELVMTQTPASLAVSLGQRATISC	23
26	ELVMTQSPASLAVSLGQSVTISCRASESVEYYGTSLMQWYQQKPGQPPKLLIYGASNVES GVPARFSGSGSGTDFSLNIHPVEEDDIAIYFCQQSRKVPATFGSGTKLEIKRADAAPTVS	ELVMTQSPASLAVSLGQSVTISC	23
27	MDFQVQIFSFLLISASVIMSRGQIVLSQSPAILSASPGEKVTMTCRASSSITYIHWYQQK SGSSPKPWIYAASNLASGVPARFSGSGSGTSYSLTISRVEAEDAATYYCQQWSRNPWTFG	QIVLSQSPAILSASPGEKVTMTC	23
28	MKLPVRLLVLMFWIPASSSDVLMTQTPLSLPVSLGDQASISCRSSLSIVHSNGNTYLEWY LQKPGQSPKLLIYKVSNRFSGVPDRFSGSGSGTDFTLKISRVEAEDLGVYYCFQGSHVPW	DVLMTQTPLSLPVSLGDQASISC	23
29	QIVLTQSPAIMSASPGEKVTMTCSASSSISYMHWYQQKPGTSPKRWIYDTSKLASGVPAR FSGSGSGTSYSLTISSMEAEDAATYYCHQRSSYPTFGAGTKLELKR	QIVLTQSPAIMSASPGEKVTMTC	23
30	DIVLTQSPASLAVSLGQRATISYRASKSVSTSGYSYMHWNQQKPGQPPRLLIYLVSNLES GVPARFSGSGSGTDFTLNIHPVEEEDAATYYCQHIRELTRSEGGPSWK	DIVLTQSPASLAVSLGQRATISY	23
31	DIELTQSPALMSASPGEKVTMTCSASSSVSYMFWYQQKPRSSPKSWIYLTSNLASGVPAR FSGSGSGTSYSLTISSMEAEDAATYYCQQWSSPLTFGAGTKLEI	DIELTQSPALMSASPGEKVTMTC	23
32	DIQMTQSPSSMYASLGERVTITCKASQDIKSYLNWYQQKPWKSPKTLIYYATSLADGVPS RFSGSGSGQDYSLTISSLESDDTATYYCLQYGESPYTFGGGTKLELKR	DIQMTQSPSSMYASLGERVTITC	23
33	QILLTQSPAIMSASPGEKVTMTCSASSSVSYMHWYQQKPGSSPKRWIYDTSKLASGVPAR FSGSGSGTSYSLIISSMEAEDAATYYCHQRSSYPFGAGTKLELKR	QILLTQSPAIMSASPGEKVTMTC	23
34	DIQMTQSPKFMSTSVGGRVNVTCKASQNVGTNVAWYQQKPGQSPKALIYSASYRYSGVPD RFTGSGSGTDFTLTISNVQSEDLAEYFCQQYNSYPFTFGSGTKLEIKR	DIQMTQSPKFMSTSVGGRVNVTC	23
35	ELVMTQSPSSLAVSAGEKVTMSCMSSQTLLNSRTRKNYLAWYQQKPGQSPTLLIYWASTR KSGVPARFTGSGSGTDFTLTISGVQAEDLAVYYCKQSYNFPTFGGGTKLEIKRADAAPTV	ELVMTQSPSSLAVSAGEKVTMSC	23
36	DIKMTQSPSSMYAFLGERVTIICKASQDIYRYLTWFQQKPGKSPKTLIYRANRLVDGGPS RFSGSGSGQDYSLTISSLEYEDMGIYYCLQYDEFPFTFGTGTKLEIK	DIKMTQSPSSMYAFLGERVTIIC	23
37	DIQMTQTTSSLSASLGDRVTISCRASQDISNYLNWYQQKPDGTVKLLIYYSSRLHSGVPS RFSGSGSGTDYSLTISNLEQEDIATYFCQQGNTLPRTFGGGTKL	DIQMTQTTSSLSASLGDRVTISC	23
38	DVVMTQTPLSLPVSLGDQASISCRSSQSIVHRNGNTYLEWYLQKPGQSPKLLIYKVSNRF SGVPDRFSGSGSGTDFTLKISRVEAEDLGVYYCFQGSHFPYTFGGGTKLELKR	DVVMTQTPLSLPVSLGDQASISC	23
39	DVLMTQTQKFMSTSVGDRVSVTCKASQNVGTNVVWYQQKPGHSPKALIYSASYRYSGVPD RFTGSGSGTDFTLTISNVQSEDLAEYFCQQYNSYPYTFGGGTKLEIE	DVLMTQTQKFMSTSVGDRVSVTC	23
40	DIVITQSPSSLTVSAGEKVTMTCKSSQSLLNSRTRKNYLAWYQQKPGQSPKLLIYWASTR ESGVPDRFTGSGSGTDFTLTISSVQAEDLAVYYCKQSYDLWTFGGGTKLEIRRADAAPTV	DIVITQSPSSLTVSAGEKVTMTC	23
41	DIVMTQSPASLSVSVGETVTITCRASENIYSNLAWYQQKQGKSPQLLVYAATNLADGVPS RFSGSGSGTQYSLKINSLQSEDFGSYYCQHFWGTPWTFGGGTKLEIKRADAAPTVS	DIVMTQSPASLSVSVGETVTITC	23
42	ETTVTQSPASLSMSIGEKVTIRCITSTDIDDDMNWYQQKPGEPPRLLISDGNTLRPGVPS RFSSSGYGTDFVFTIENMLSEDVADYYCLQSDNLPYTFGGGTNLEIK	ETTVTQSPASLSMSIGEKVTIRC	23
43	DIVLTQSPPSLAVSLGQRATISCRASESIDLYGFTFMHWYQQKPGQPPKILIYRASNLES GIPARFSGSGSRTDFTLTINPVEADDVATYYCQQTHEDPYTFGGGTKLEIK	DIVLTQSPPSLAVSLGQRATISC	23
44	QIVLTQSPAIMSASPGEKVTMTCSASSSVSYMYWYQQKPGSSPRLLIYDTSNLASGVPVR FSGSGSGTSYFLTISRMEAEDAATYYCQQWSSYP	QIVLTQSPAIMSASPGEKVTMTC	23
45	DIVMTQSHKFMSTSVGDRVSVTCKASQNVGTNVAWYQQKPGQSPKALIYWASNRFTGVPD RFTGSGSGTDFTLTISNVQSEDLADYFCQQYSSSPFTFGTGTKLEIK	DIVMTQSHKFMSTSVGDRVSVTC	23
46	QILLTQSPAFMSASPGEKVTMTCSASSSVSYIHWYQQKPGSSPKPWIYDTSNLASGFPAR CSGSGSGTSYSLIISSMEAEDAATYYCHQRSSYPLTFGAGTKLELKR	QILLTQSPAFMSASPGEKVTMTC	23
47	DVVMTQTQKLMSTSVGDRVSVTCKASQNVGTNVAWYQQKPGQSPKALIYSASYRYSGVPD RFTGSGSGTDFTLTISNVQSEDLTEYFCQQYNGYPLTFGAGTKLEIK	DVVMTQTQKLMSTSVGDRVSVTC	23
48	DVLMTQTPXSLPVSLGDQASISCRSSQSIVXSNGNTYLEWYLQKPGQSPKLLIYKVSNRF SGVPDRFSGSGSGXDFTLKISRVEXXDLGVYYCFQGSHVPWTFGGGTKLEIK	DVLMTQTPXSLPVSLGDQASISC	23
49	MVFTPQILGLMLFWISASRGDIVLTQSPATLSVTPGDSVSLSCRASQSISNNLHWYQQKS HESPRLLIKYASQSISGIPSRFSGSGSGTDFTLSINSVETEDFGMYFCQQSNSWPFTFGS	DIVLTQSPATLSVTPGDSVSLSC	23
50	QIVLTQSPAIMSASPGEKVTMTCSASSSISYMHWYQQKPGTSPKRWIYDTSKLASGVPAR FSGSGSGTSYSLTISSMEAEDAATYYCHQRSSYPTFGAGTKLELKR	QIVLTQSPAIMSASPGEKVTMTC	23

```
# import file into aFile and then into aList
aFile = open('FR1_and_complete.txt', 'r')
aList = aFile.readlines()

# eliminate first four lines of junk at the beginning of file
del aList[0:4]

print("The total number of sequences in the database is:")
print(len(aList))
print(' ')

# create aGoodList with placeholders
aGoodList = [('start','start','start'), ('end','end','end')]
for i in range(len(aList)):
    aGoodList[(i):(i+1)] = [('0','0','0')]

# Populate aGoodList with the first two blocks of sequence and then
the FR1 region
for i in range(len(aList)):
    Item = aList[i]
    ItemMod = Item.split()
    aGoodList[i] = (ItemMod[1],ItemMod[2],ItemMod[-2])

# Use the set function to remove duplicates
aBetterList = list(set(aGoodList))

print("After removing duplicate sequences, there are:")
print(len(aBetterList))
print(' ')

# make new list for just the FR1 sequence
aNewList = ['start','end']
for i in range(len(aBetterList)):
    aNewList[(i):(i+1)] = ['0']

# Populate aNewList with the FR1 region sequence
for i in range(len(aBetterList)):
    Item = aBetterList[i]
    aNewList[i] = (Item[2])

# make new list with just the first 7 amino acids and eliminate any
entries that contain an X
anImprovedList = [i[0:7] for i in aNewList]
aBestList = [i for i in anImprovedList if not 'X' in i]

print("After removing all of the incomplete sequences (with an X),
there are:")
print(len(aBestList))
print(' ')

# count unique 7 amino acid FR1 entries and # of occurences
from collections import Counter
Uniques = Counter(aBestList)
print("The number of unique 7 amino acid FR1 sequences in these 2237
mAbs:")
print(len(Uniques))
print(' ')

# Statistics
UniqueList = list(Uniques.items())
UniqueList.sort(key=lambda x: int(x[1]))
UniqueList.reverse()
print("The top 15 most frequent sequences, and their frequencies,
are:")
print(' ')
print(UniqueList[0:15])
Top15 = UniqueList[0:15]
Top15Nums = [int(x[1]) for x in Top15]
Top15Sum = sum(Top15Nums)
Top15Percent = (Top15Sum/len(aBestList))*100
print(' ')
print('Of the 2237 mAbs, the top 15 sequences account for:')
print(Top15Sum)
```

```
Python 2.7.6 (default, Sep  9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ==================== RESTART ==================
>>>
The total number of sequences in the database is:
2676

After removing duplicate sequences, there are:
2265

After removing all of the incomplete sequences (with an X), there are:
2237

The number of unique 7 amino acid FR1 sequences in these 2237 mAbs:
190

The top 15 most frequent sequences, and their frequencies, are:

[('QIVLTQS', 259), ('DIVMTQS', 231), ('DIVLTQS', 181), ('DIQLTQS',
169), ('DVVMTQT', 163), ('DIQMTQS', 155), ('DVLMTQT', 115),
('DIQMTQT', 107), ('DIELTQS', 64), ('ELVMTQS', 51), ('DIVMSQS', 50),
('EIVLTQS', 49), ('ENVLTQS', 42), ('DIKMTQS', 41), ('DIVMTQA', 37)]

Of the 2237 mAbs, the top 15 sequences account for:
1714
>>>
```

```
# import file into aFile and then into aList
aFile = open('FR1_and_complete.txt', 'r')
aList = aFile.readlines()

# eliminate first four lines of junk at the beginning of file
del aList[0:4]

print("The total number of sequences in the database is:")
print(len(aList))
print(' ')

# create anSPList with placeholders
anSPList = [('start','start','start'), ('end','end','end')]
for i in range(len(aList)):
    anSPList[(i):(i+1)] = [('0','0','0')]

# Populate anSPList with two blocks of sequence and then the 1st 10 of
SP region
for i in range(len(aList)):
    Item = aList[i]
    ItemMod = Item.split()
    S = ItemMod[1]
    SP = S[:10]
    if S[0] == 'M':
        anSPList[i] = (ItemMod[1], ItemMod[2], SP)

# and now we remove all 0's from anSPList
aGoodSPList = [i for i in anSPList if i != ('0','0','0')]

print("The total number of light chains containing SP sequence is:")
print(len(aGoodSPList))
print(' ')

# Use the set function to remove duplicates
aBetterSPList = list(set(aGoodSPList))

# make new list for just the SP sequence
aNewList = ['start','end']
for i in range(len(aBetterSPList)):
    aNewList[(i):(i+1)] = ['0']

# Populate aNewList with just the SP region sequence
for i in range(len(aBetterSPList)):
    Item = aBetterSPList[i]
    aNewList[i] = (Item[2])

# eliminate any entries that contain an X
aBestList = [i for i in aNewList if not 'X' in i]

print("After removing duplicate and incomplete sequences (with an X),
there are:")
print(len(aBestList))
print(' ')

# count unique 10 amino acid SP entries and # of occurences
from collections import Counter
Uniques = Counter(aBestList)
print("The number of unique 10 amino acid SP sequences in these 449
mAbs:")
print(len(Uniques))
print(' ')

# Statistics
UniqueList = list(Uniques.items())
UniqueList.sort(key=lambda x: int(x[1]))
UniqueList.reverse()
print("The top 15 most frequent sequences, and their frequencies,
are:")
print(' ')
print(UniqueList[0:15])
Top15 = UniqueList[0:15]
Top15Nums = [int(x[1]) for x in Top15]
Top15Sum = sum(Top15Nums)
Top15Percent = (Top15Sum/len(aBestList))*100
print(' ')
print('Of the 449 mAbs, the top 15 sequences account for:')
print(Top15Sum)
```

```
Python 2.7.6 (default, Sep  9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART
================================
>>>
The total number of sequences in the database is:
2676

The total number of light chains containing SP sequence is:
524

After removing duplicate and incomplete sequences (with an X), there
are:
449

The number of unique 10 amino acid SP sequences in these 449 mAbs:
141

The top 15 most frequent sequences, and their frequencies, are:

[('MDFQVQIFSF', 67), ('MKLPVRLLVL', 67), ('METDTLLLWV', 20),
('MMSSAQFLGL', 9), ('MSVPTQVLGL', 9), ('MDSQAQVLML', 7),
('MDMRAPAQIF', 7), ('METDTILLWV', 7), ('MVFTPQILGL', 7),
('MMSPAQFLFL', 6), ('MESQTQVLMF', 5), ('MRCLAEFLGL', 5),
('MKSQTQVFIF', 5), ('MSVLTQVLAL', 5), ('METHSQVFVY', 5)]

Of the 449 mAbs, the top 15 sequences account for:
231
>>>
```



Supplemental Figure 2. **Binding of scFv-decorated liposomes to HUVEC.** (a) Ab62 scFv decorated fluoroliposomes bound to HUVEC after 30 minute incubation under static conditions. (b) Mec13 scFv decorated liposomes were used as a control for non-specific binding.