In [144]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
```

In [145]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=Tru
e).
```

In [146]:
```
errors = pd.read_csv("/content/drive/My Drive/cm121/errors.tsv", sep='\t', header=None)
errors
```

Out[146]:

	0
0	0.002000
1	0.003996
2	0.005988
3	0.007976
4	0.009960
...	...
95	0.174852
96	0.176502
97	0.178149
98	0.179793
99	0.181433

100 rows × 1 columns

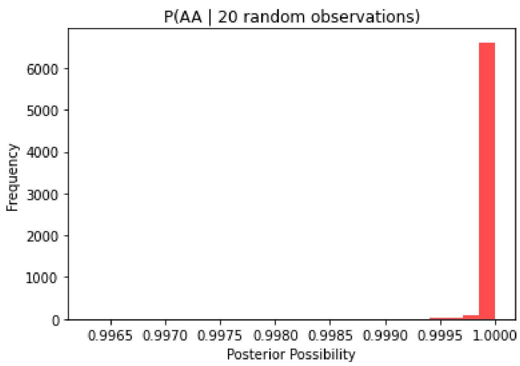
In [147]:
```
transitions = pd.read_csv("/content/drive/My Drive/cm121/transitions.tsv", sep='\t', header=None)
# A C G T
transitions
```

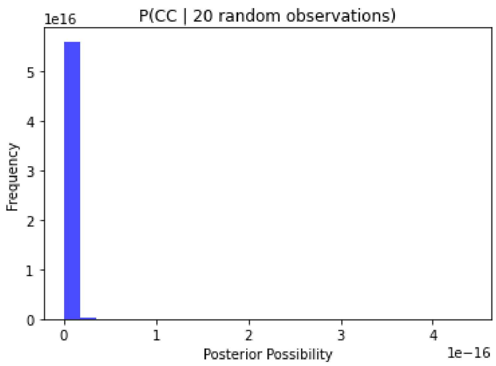
Out[147]:

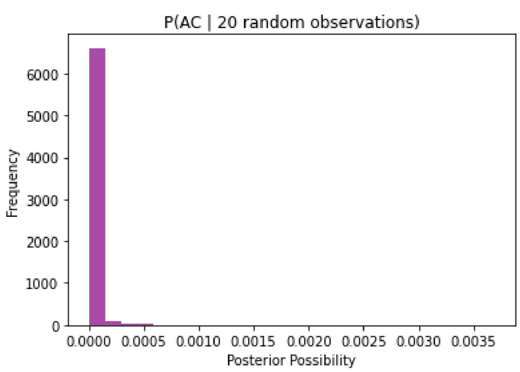
	0	1	2	3
0	0.0	0.2	0.2	0.6
1	0.3	0.0	0.6	0.1
2	0.2	0.7	0.0	0.1
3	0.5	0.3	0.2	0.0

In [148]:
```
## part c
aa_hist = []
cc_hist = []
ac_hist = []
for i in range(1000):
  data = pd.DataFrame({"obs":[], "p_error":[]})
  n_error = 0
  # true genotype is AA
  N = 20
  for _ in range(N):
      obs = 'A'
      rand_err = random.randint(0, 99)
      e = errors.iloc[rand_err][0]
      if random.random() < e:
          n_error += 1
          p_transition = random.randint(1,100)
          if p_transition <= 20:
              obs = 'C'
          elif p_transition <= 40:
              obs = 'G'
          else:
              obs = 'T'

      data.loc[len(data.index)] = [obs, e]
  data['p_truth'] = 1 - data['p_error']

  AA = []
  for i in range(N):
    if data["obs"][i] == "A":
      AA.append(data["p_truth"][i])
    else:
      AA.append(data["p_error"][i])

  CC = []
  for i in range(N):
    if data["obs"][i] == "C":
      CC.append(data["p_truth"][i])
    else:
      CC.append(data["p_error"][i])

  AC = []
  for i in range(N):
    if data["obs"][i] == "A":
      AC.append(0.5 * data["p_truth"][i] + 0.5 * data["p_error"][i])
    elif data["obs"][i] == "C":
      AC.append(0.5 * data["p_truth"][i] + 0.5 * data["p_error"][i])
    else:
      AC.append(data["p_error"][i])

  p_data_given_AA = np.prod(AA)
  p_data_given_CC = np.prod(CC)
  p_data_given_AC = np.prod(AC)

  p_AA = 0.95**2
  p_CC = 0.05**2
  p_AC = 1 - p_AA - p_CC

  p_data_and_AA = p_data_given_AA * p_AA
  p_data_and_CC = p_data_given_CC * p_CC
  p_data_and_AC = p_data_given_AC * p_AC

  p_data = p_data_and_AA + p_data_and_CC + p_data_and_AC

  aa_hist.append(p_data_and_AA / p_data)
  cc_hist.append(p_data_and_CC / p_data)
  ac_hist.append(p_data_and_AC / p_data)
```

In [149]:
```
plt.hist(aa_hist, 25, density = 1, color ='red', alpha = 0.7)
plt.xlabel('Posterior Possibility')
plt.ylabel('Frequency')

plt.title('P(AA | 20 random observations)')
plt.show()

plt.hist(cc_hist, 25, density = 1, color ='blue', alpha = 0.7)
plt.xlabel('Posterior Possibility')
plt.ylabel('Frequency')

plt.title('P(CC | 20 random observations)')
plt.show()

plt.hist(ac_hist, 25, density = 1, color ='purple', alpha = 0.7)
plt.xlabel('Posterior Possibility')
plt.ylabel('Frequency')

plt.title('P(AC | 20 random observations)')
plt.show()

print("P(AA | data):", np.mean(aa_hist))
print("P(CC | data):", np.mean(cc_hist))
print("P(AC | data):", np.mean(ac_hist))
```



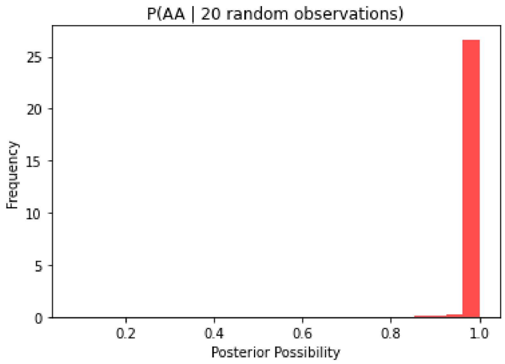


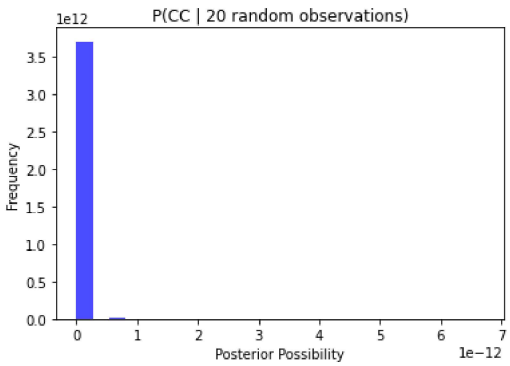


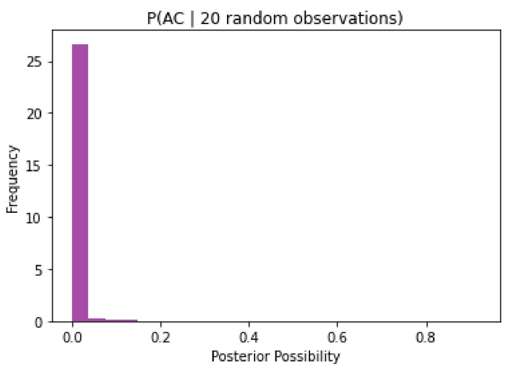

```
P(AA | data): 0.9999803716098639
P(CC | data): 1.4807538228830768e-18
P(AC | data): 1.9628390136120132e-05
```

In [150]:
```
## part d
aa_hist = []
cc_hist = []
ac_hist = []
for i in range(1000):
  new_data = pd.DataFrame({"obs":[], "p_error":[]})

  n_error = 0
  # true genotype is AA
  N = 20
  for _ in range(N):
      obs = 'A'
      rand_err = random.randint(0, 99)
      e = errors.iloc[rand_err][0]
      if random.random() < e: # if error
          obs = 'C'
      new_data.loc[len(new_data.index)] = [obs, e]
  new_data['p_truth'] = 1 - data['p_error']

  AA = []
  for i in range(N):
    if new_data["obs"][i] == "A":
      AA.append(new_data["p_truth"][i])
    else:
      AA.append(new_data["p_error"][i])

  CC = []
  for i in range(N):
    if new_data["obs"][i] == "C":
      CC.append(new_data["p_truth"][i])
    else:
      CC.append(new_data["p_error"][i])

  AC = []
  for i in range(N):
    if new_data["obs"][i] == "A":
      AC.append(0.5 * new_data["p_truth"][i] + 0.5 * new_data["p_error"][i])
    else:
      AC.append(0.5 * new_data["p_truth"][i] + 0.5 * new_data["p_error"][i])

  p_data_given_AA = np.prod(AA)
  p_data_given_CC = np.prod(CC)
  p_data_given_AC = np.prod(AC)

  p_AA = 0.95**2
  p_CC = 0.05**2
  p_AC = 1 - p_AA - p_CC

  p_data_and_AA = p_data_given_AA * p_AA
  p_data_and_CC = p_data_given_CC * p_CC
  p_data_and_AC = p_data_given_AC * p_AC

  p_data = p_data_and_AA + p_data_and_CC + p_data_and_AC

  aa_hist.append(p_data_and_AA / p_data)
  cc_hist.append(p_data_and_CC / p_data)
  ac_hist.append(p_data_and_AC / p_data)
```

In [151]:
```
plt.hist(aa_hist, 25, density = 1, color ='red', alpha = 0.7)
plt.xlabel('Posterior Possibility')
plt.ylabel('Frequency')

plt.title('P(AA | 20 random observations)')
plt.show()

plt.hist(cc_hist, 25, density = 1, color ='blue', alpha = 0.7)
plt.xlabel('Posterior Possibility')
plt.ylabel('Frequency')

plt.title('P(CC | 20 random observations)')
plt.show()

plt.hist(ac_hist, 25, density = 1, color ='purple', alpha = 0.7)
plt.xlabel('Posterior Possibility')
plt.ylabel('Frequency')

plt.title('P(AC | 20 random observations)')
plt.show()

print("P(AA | data):", np.mean(aa_hist))
print("P(CC | data):", np.mean(cc_hist))
print("P(AC | data):", np.mean(ac_hist))
```








```
P(AA | data): 0.9953283788191177
P(CC | data): 1.7010642411528735e-14
P(AC | data): 0.004671621180865315
```

In [151]: