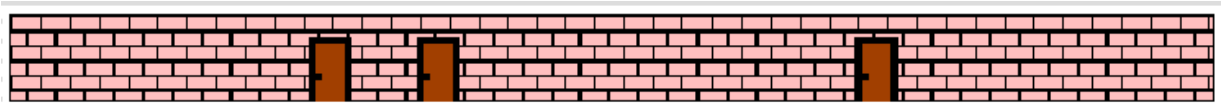
In [3]:
```
'''
Created on Jun 3, 2012

Localizacao Monte Carlo - Filtro de Particulas

@author: Rafael Luiz Klaser

uso:
- o metodo monte_carlo_filter() executa o metodo em um unico passo
- os metodos passo a passo sao:
    step_move()     : gera um deslocamento (positivo ou negativo)
    step_sense()    : executa uma medicao (ex: viu porta)
    step_resample() : reposiciona as particulas de acordo com a crenca gerada por sense

ex execucao:
    step_move(2.4)
    step_sense('porta')
    step_resample()

- o metodo show_filter() gera um grafico 'scatter' exibindo a distribuicao
  das particulas em X
  ex:
      executar show_filter() a cada passo para ver as particulas
'''

import random
import matplotlib.pyplot as plt

mundo=['par', 'porta', 'par', 'porta', 'par', 'par', 'par', 'porta', 'par', 'par']

#Monte Carlo
random.seed()

#constantes
min_ambiente = 0.0
max_ambiente = 10.0
num_particulas = 200
base_weight = 1.0/num_particulas

#ruido do deslocamento
max_ruido = 0.17
min_ruido = 0.03

#particulas
_Xt=[0]*num_particulas
_Wt=[0]*num_particulas


'''
Algoritmo MCL
'''
#- Ut -> Odometria (comando movimento)
#- Zt -> Sensores (observacao)
def monte_carlo_filter(Ut, Zt):
    for m in range(num_particulas):
        _Xt[m] = sample_motion_model(Ut, _Xt[m])
        _Wt[m] = measurement_model(Zt, _Xt[m], m)

    _Xt = resample()
    _Wt = [base_weight]*num_particulas

'''
Execucoes passo-a-passo
'''
def step_sense(Zt):
    for m in range(num_particulas):
        _Wt[m] = measurement_model(Zt, _Xt[m], m)

def step_move(Ut):
    for m in range(num_particulas):
        _Xt[m] = sample_motion_model(Ut, _Xt[m])
```

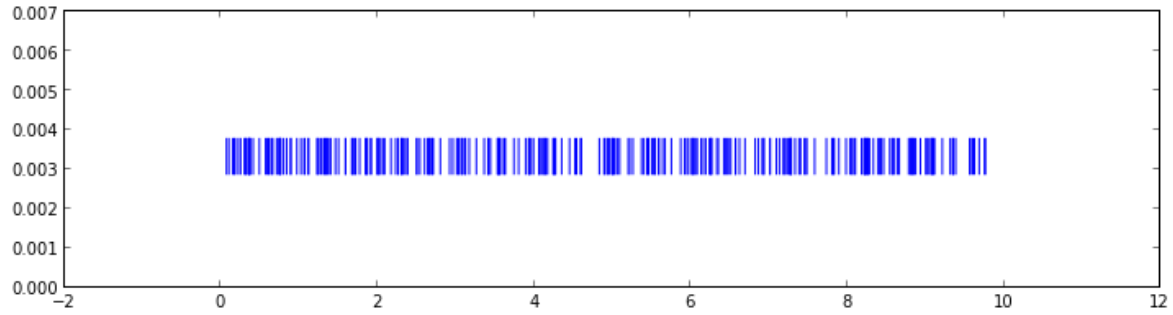
```
def step_resample():
    Xt = resample()
    for m in range(num_particulas):
        _Xt[m] = Xt[m]
        _Wt[m] = base_weight

def new_particle():
    #gera particulas com 2 casas decimais
    return random.randint(min_ambiente*100, max_ambiente*99) / 100.0

#- distribui as particulas aleatoreamente pelo espaco
#- inicializa distribuicao uniforme
def initialize():
    for i in range(num_particulas):
        _Xt[i] = new_particle()
        _Wt[i] = base_weight

def normalize():
    norm=sum(_Wt)
    for j in range(num_particulas):
        _Wt[j]=_Wt[j]/norm

def resample():
    '''
    usa os vetores _Xt e _Wt como sendo a p(z|x) X(barra)t
    '''
    #normaliza para fazer o sorteio
    normalize()

    #novas particulas
    Xt=[(0)]*num_particulas

    for m in range(num_particulas):
        i = draw_with_probability()
        Xt[m]=_Xt[i]

    return Xt

def exec_filter(Ut, Zt):
    monte_carlo_filter(Ut, Zt)

def show_exec_filter(Ut, Zt):
    exec_filter(Ut, Zt)
    show_filter()

def show_filter():
    plt.figure(figsize=(12,3))
    plt.scatter(_Xt, _Wt, s=500, marker='|')
    plt.draw()

#- retorna um indice baseado no sorteio dentro da
#distribuicao de probabilidade (roleta)
def draw_with_probability():
    roleta = random.random()
    total = 0.0
    ndx = 0
    for i in range(num_particulas):
        total+=_Wt[i]
        if(total>=roleta):
            ndx = i
            break

    return ndx

def noise(value):
    sgn = pow(-1, random.randint(1, 2))
    #gera ruido dentro do intervalo
    n = random.randint(int(min_ruido*100), int(max_ruido*100)) / 100.0
    value += (value*n/100) * sgn
    return value
```

```
#calcula nova posicao dado o comando de deslocamento
def sample_motion_model(Ut, Xt):
    #atua
    motion_pos = Xt+Ut
    #adiciona ruido
    motion_pos=noise(motion_pos)

    #valida particula
    #if(motion_pos>=max_ambiente or motion_pos<min_ambiente):
    #    #saiu do ambiente, gera nova particula aleatoria
    #    motion_pos = new_particle()

    #cilcula no ambiente
    if(motion_pos>max_ambiente):
        motion_pos = motion_pos - max_ambiente
    if(motion_pos<min_ambiente):
        motion_pos = motion_pos + max_ambiente

    return motion_pos

#medicao
def measurement_model(Zt, Xt, m):
    ndx = int(Xt)
    seen = mundo[ndx]
    norm = Xt-ndx
    hit = 0.5
    if(seen==Zt):
        #TODO: correto seria usar a curva gaussiana
        if(norm>=0 and norm<0.1):
            hit=1.20
        if(norm>=0.1 and norm<0.2):
            hit=1.33
        if(norm>=0.2 and norm<0.3):
            hit=1.45
        if(norm>=0.3 and norm<0.4):
            hit=1.70
        if(norm>=0.4 and norm<0.6):
            hit=2.00
        if(norm>=0.6 and norm<0.7):
            hit=1.70
        if(norm>=0.7 and norm<0.8):
            hit=1.45
        if(norm>=0.8 and norm<0.9):
            hit=1.33
        if(norm>=0.9 and norm<=1):
            hit=1.20

    return base_weight*hit


initialize()

from IPython.core.display import Image
Image("https://dl.dropbox.com/u/62929183/portas-pacman.png")
```

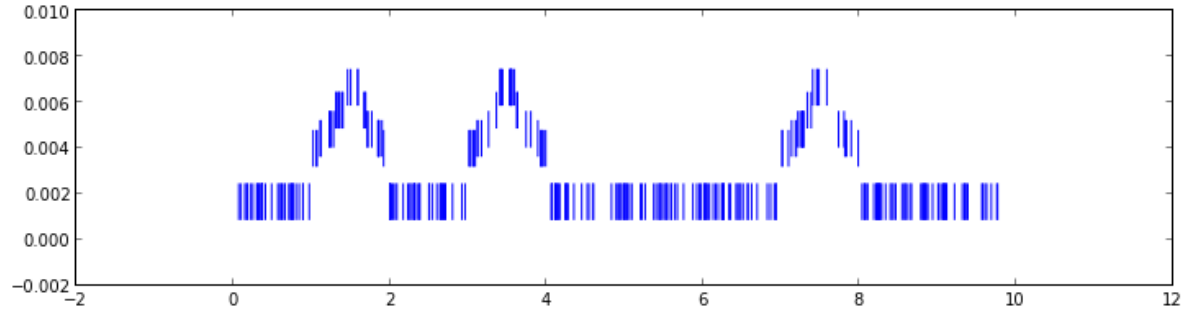
Out[3]:



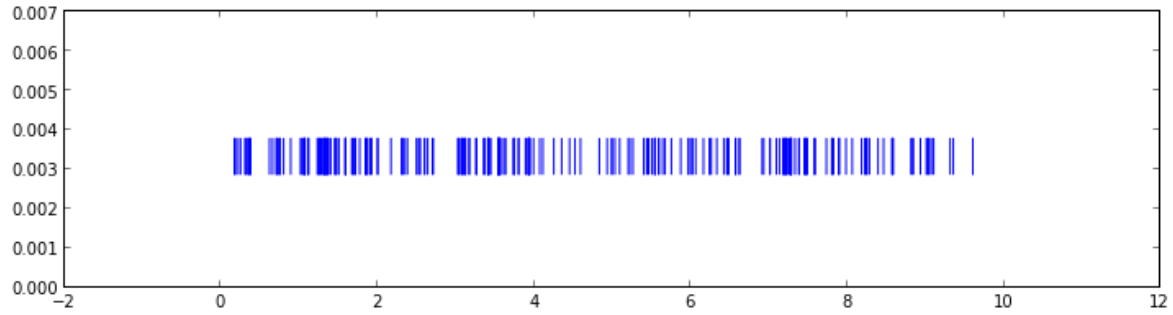
In [145]:
```
show_filter()
```



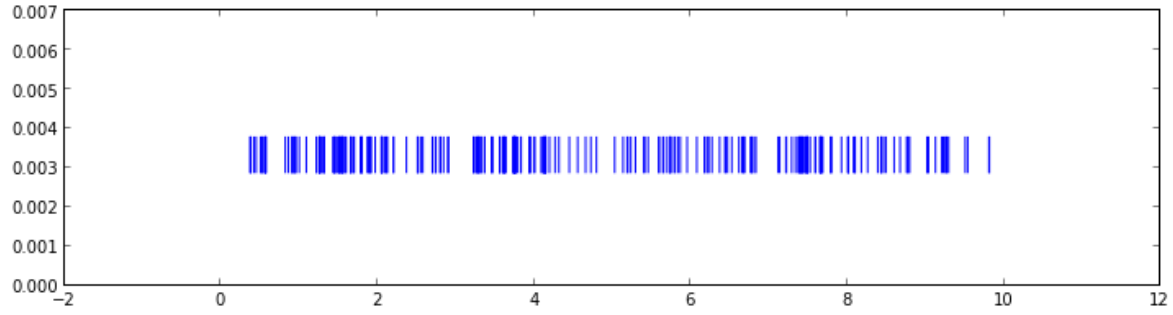
In [146]:
```
step_sense('porta')
show_filter()
```



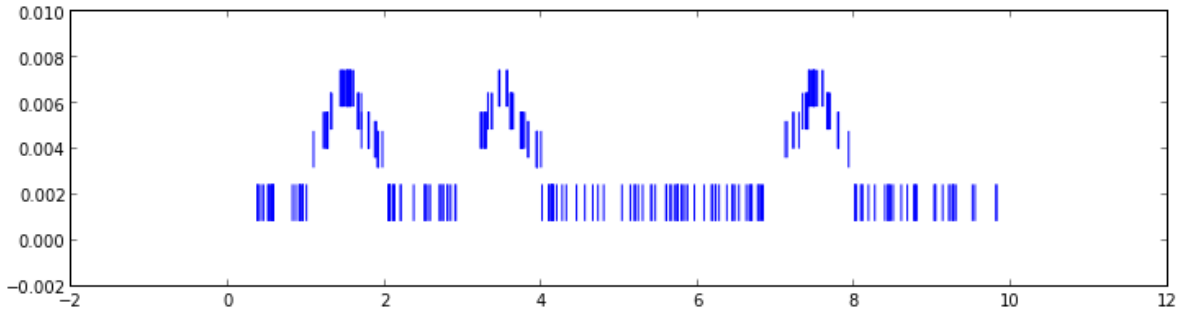
In [147]:
```
step_resample()
show_filter()
```



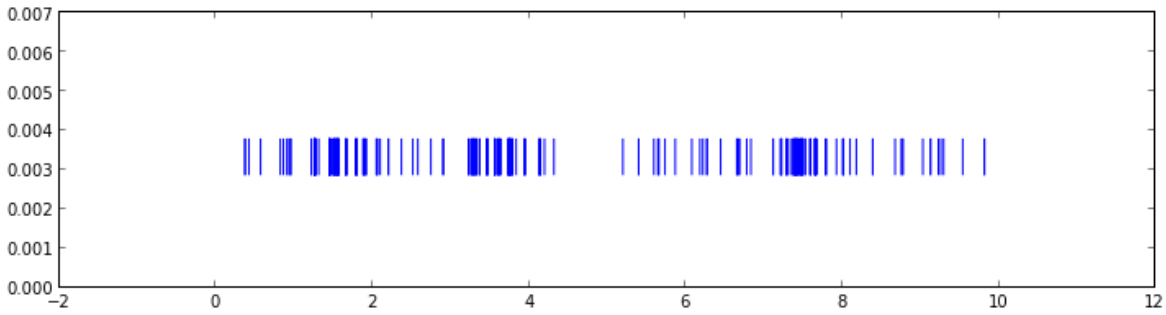
In [148]:
```
step_move(0.2)
show_filter()
```



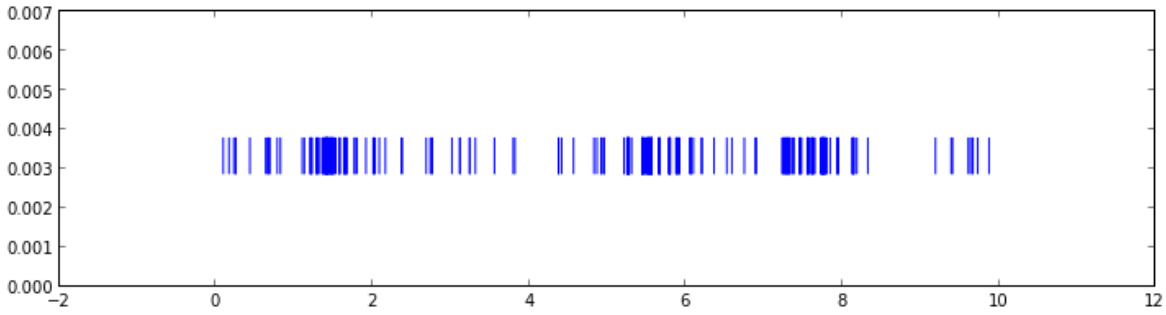
In [149]:
```
step_sense('porta')
show_filter()
```



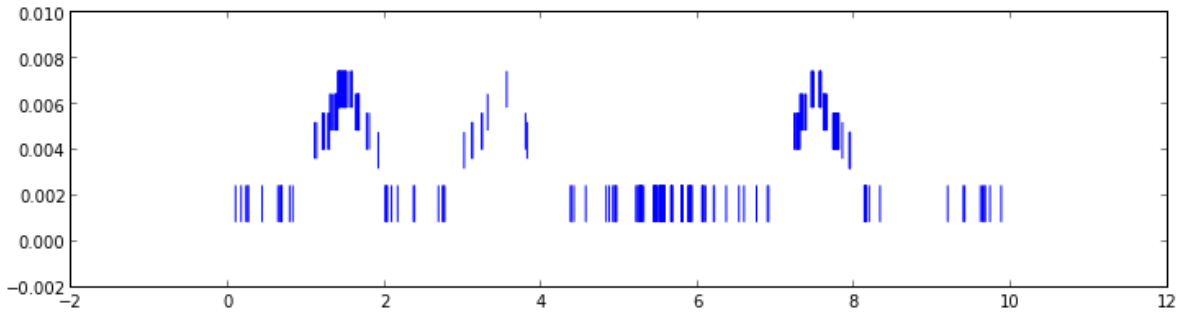
In [150]:
```
step_resample()
show_filter()
```



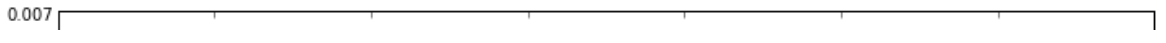
In [151]:
```
step_move(4)
show_filter()
```

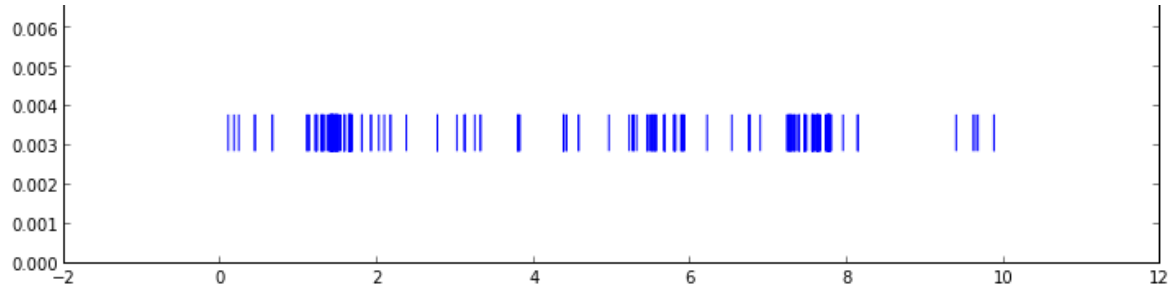


In [152]:
```
step_sense('porta')
show_filter()
```

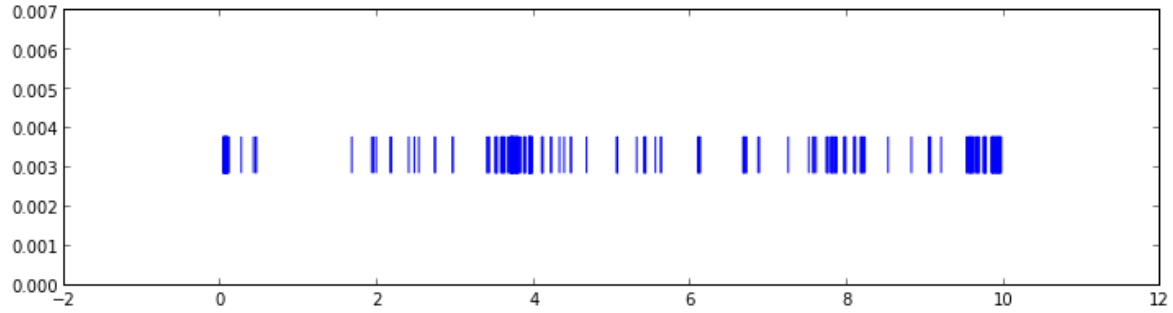


In [153]:
```
step_resample()
show_filter()
```

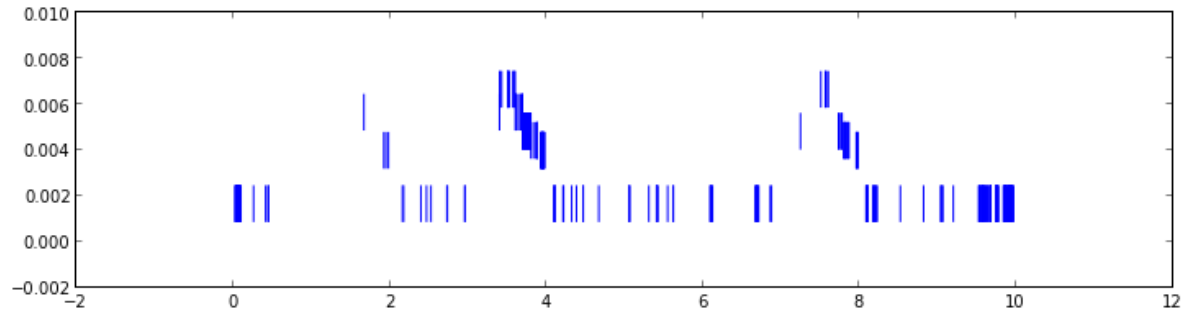




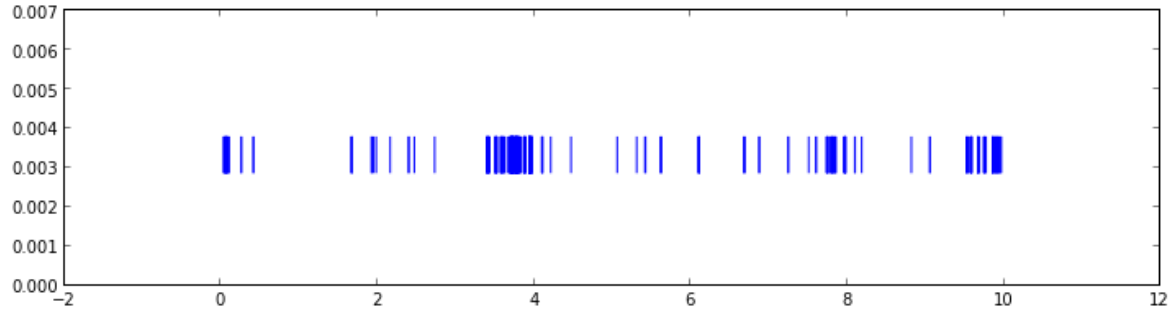
In [154]:
```
step_move(2.3)
show_filter()
```



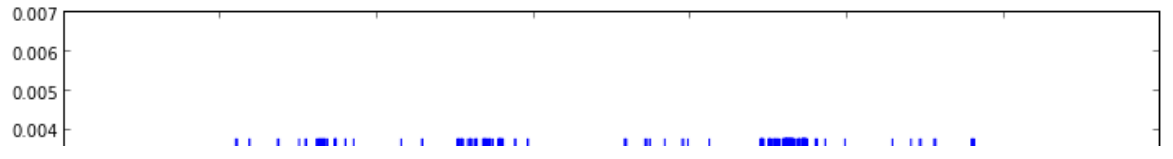
In [155]:
```
step_sense('porta')
show_filter()
```

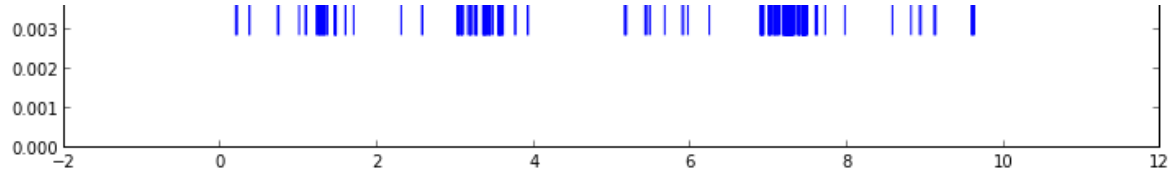


In [156]:
```
step_resample()
show_filter()
```

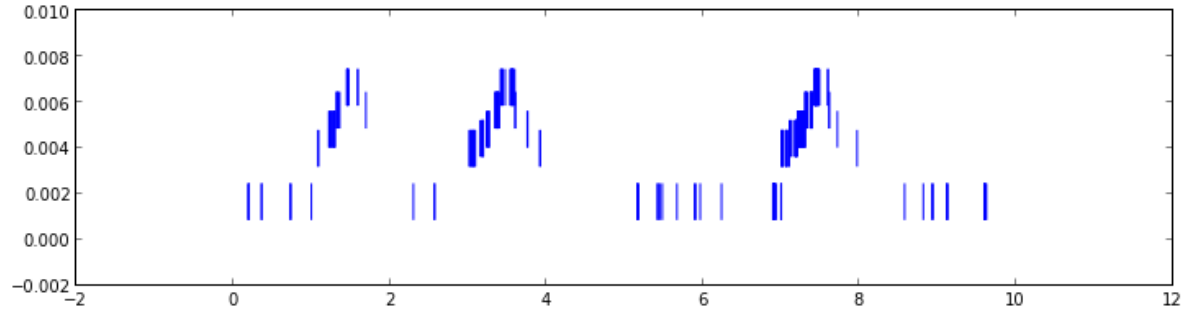


In [157]:
```
step_move(3.5)
show_filter()
```

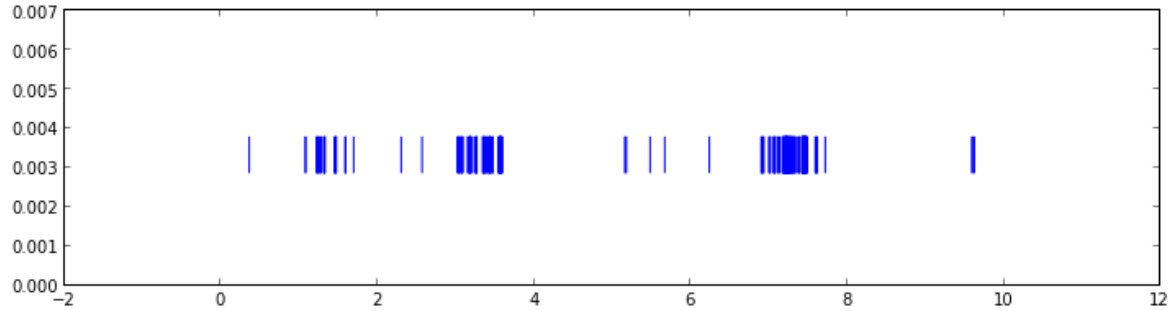




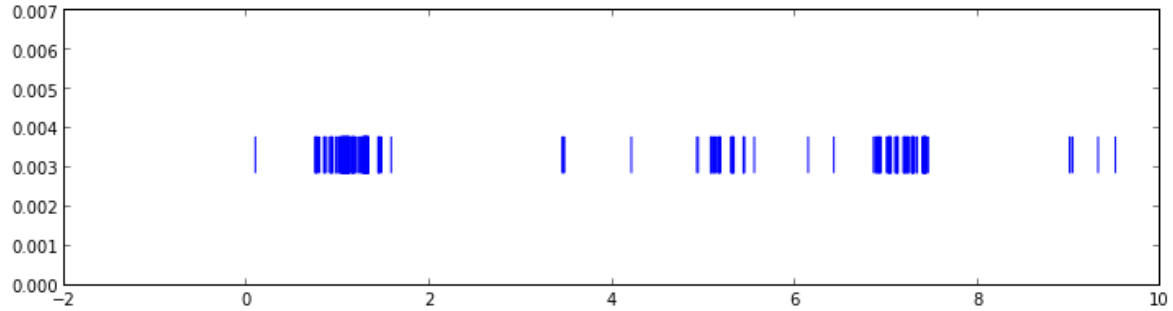
In [158]:
```
step_sense('porta')
show_filter()
```



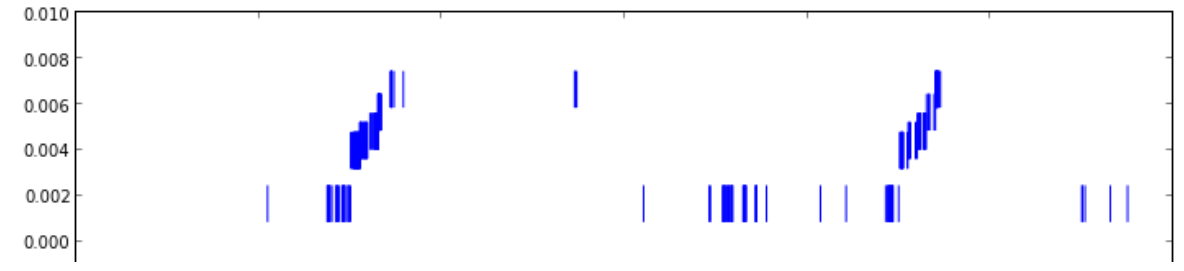
In [159]:
```
step_resample()
show_filter()
```

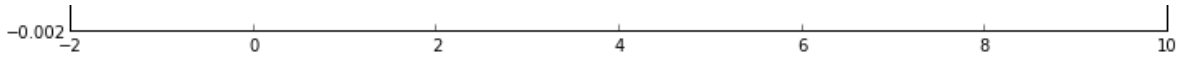


In [160]:
```
step_move(3.85)
show_filter()
```

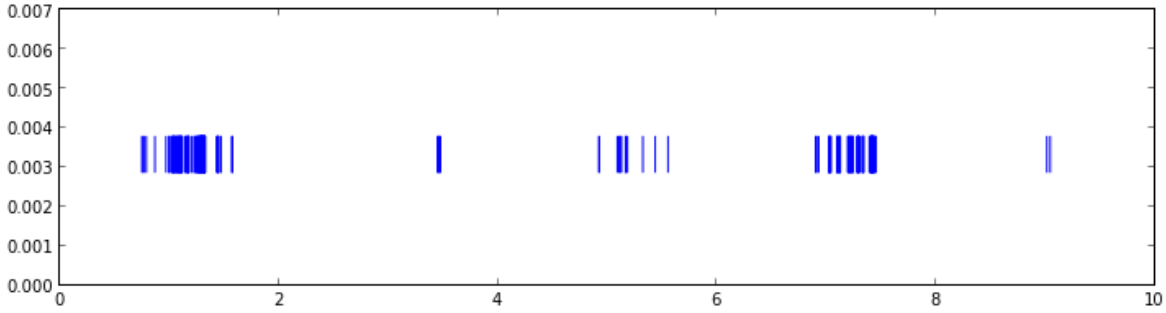


In [161]:
```
step_sense('porta')
show_filter()
```

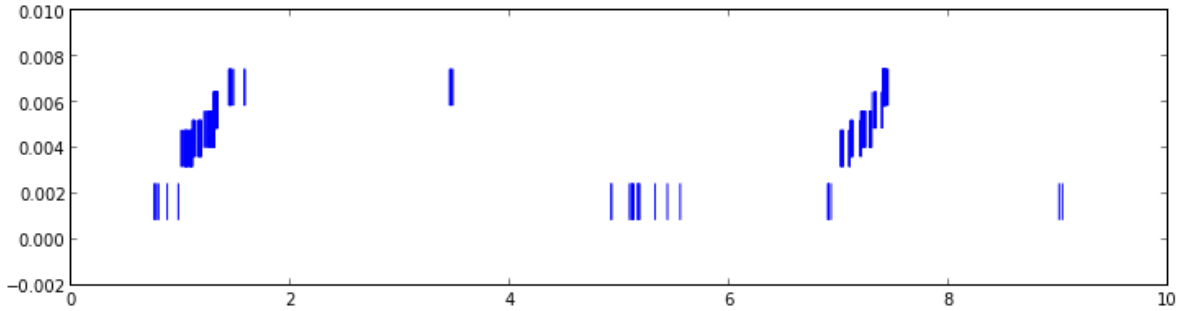




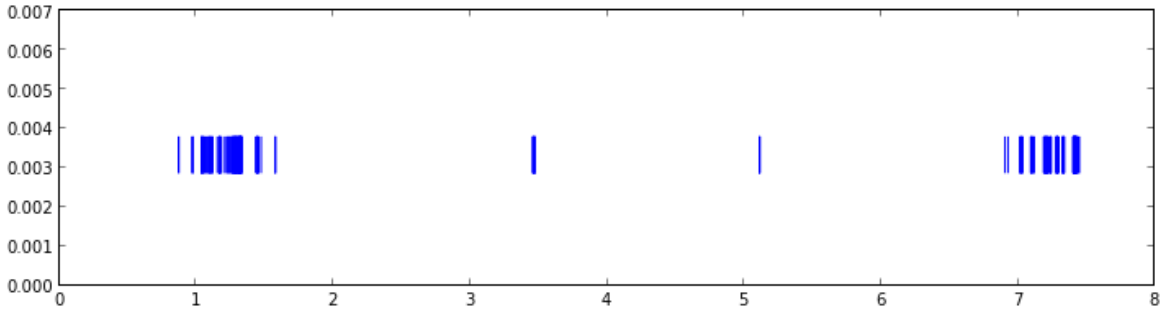
In [162]:
```
step_resample()
show_filter()
```



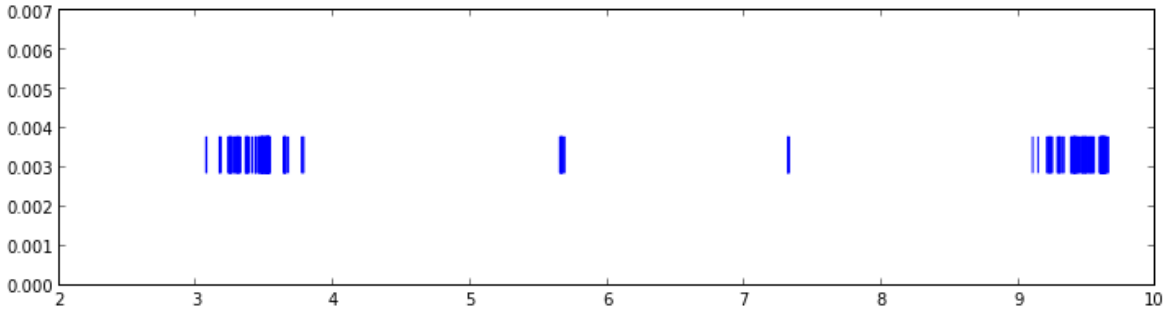
In [163]:
```
step_sense('porta')
show_filter()
```



In [165]:
```
step_resample()
show_filter()
```

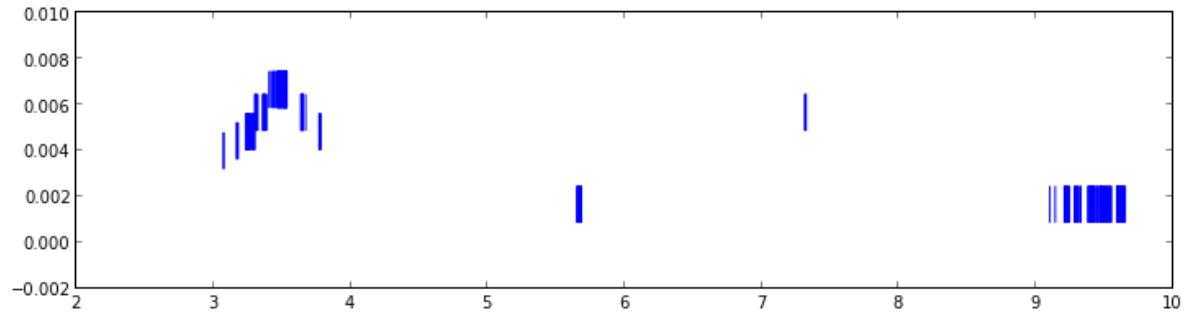


In [166]:
```
step_move(2.2)
show_filter()
```

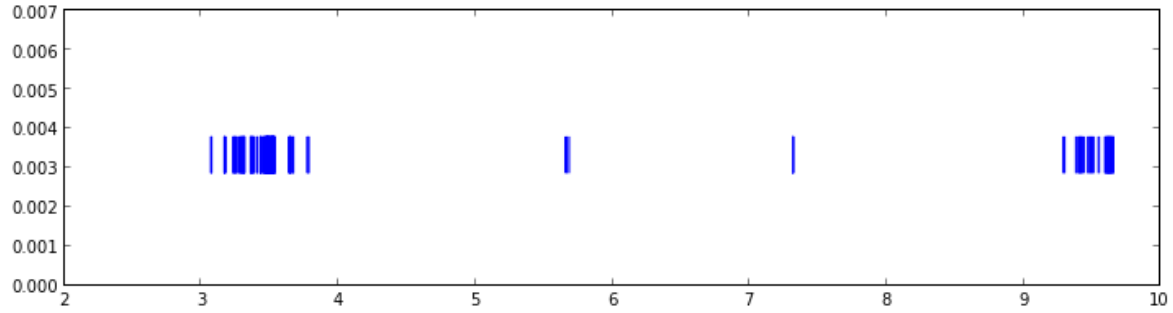


In [167]:
```
step_sense('porta')
```

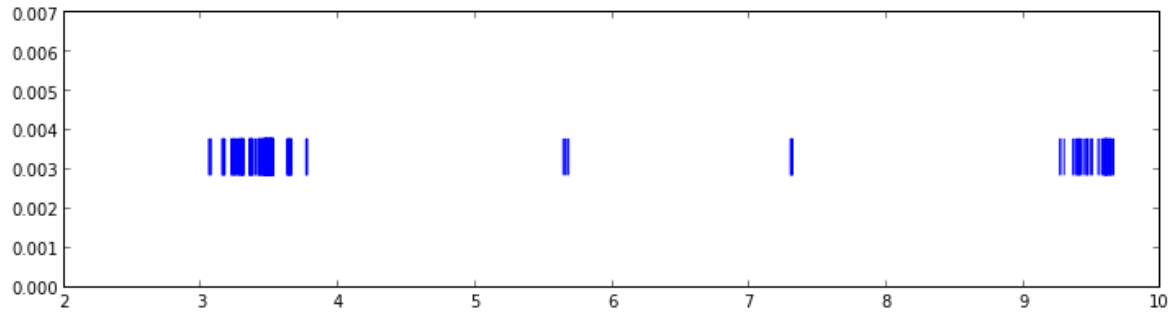
```
show_filter()
```



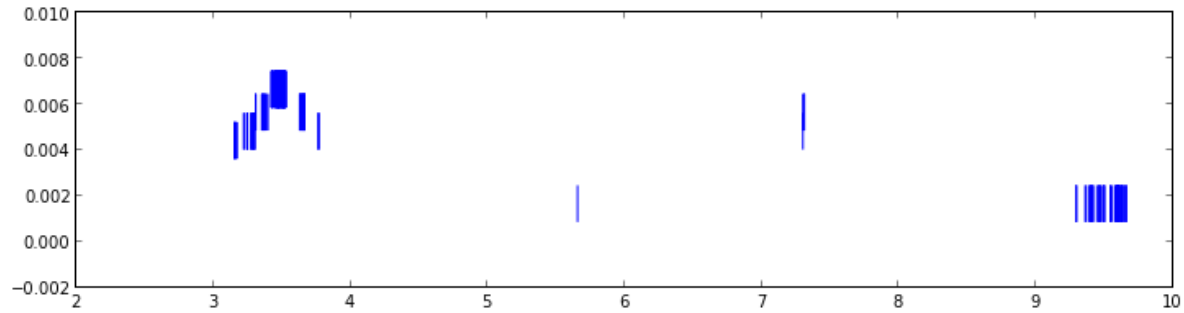
In [168]:
```
step_resample()
show_filter()
```



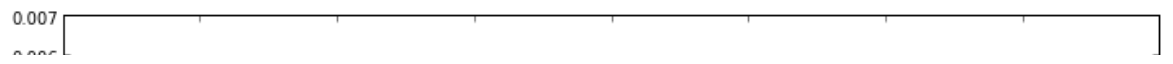
In [169]:
```
step_move(-0.01)
show_filter()
```

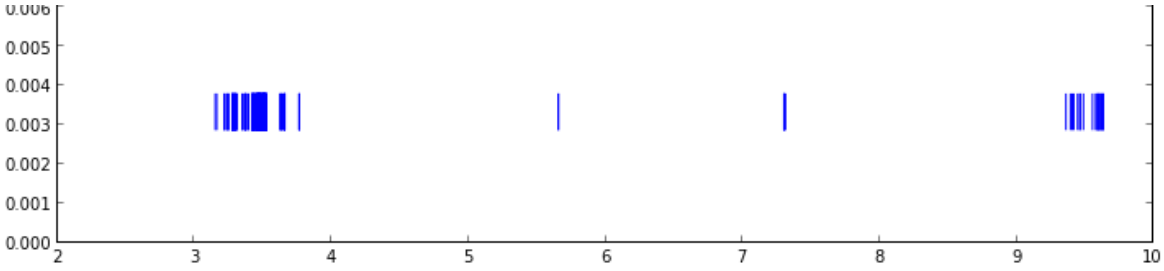


In [171]:
```
step_sense('porta')
show_filter()
```

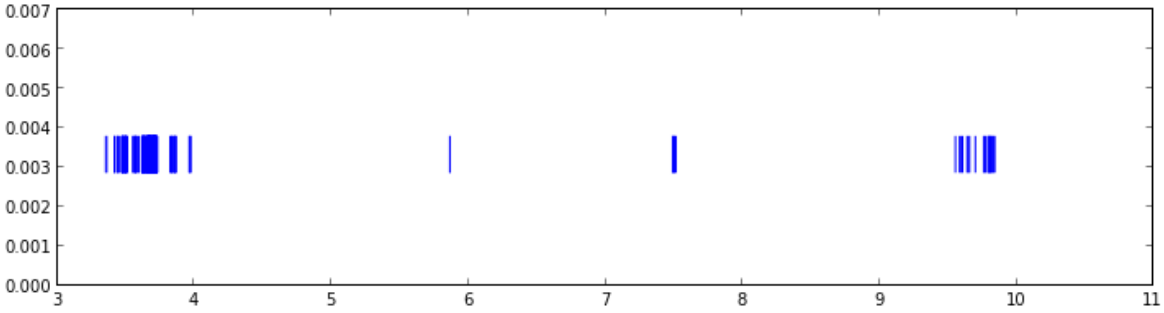


In [172]:
```
step_resample()
show_filter()
```

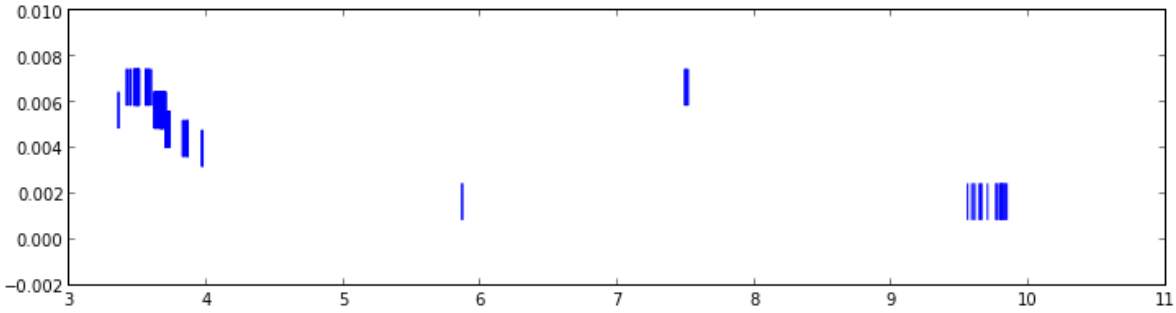




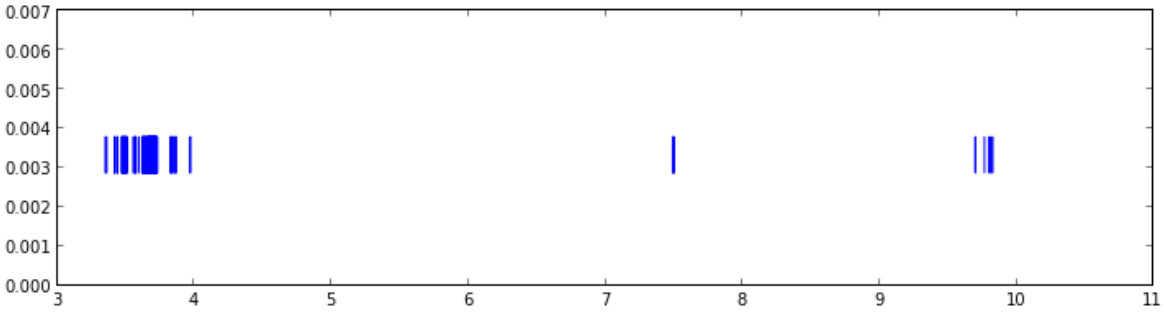
In [173]:
```
step_move(0.2)
show_filter()
```



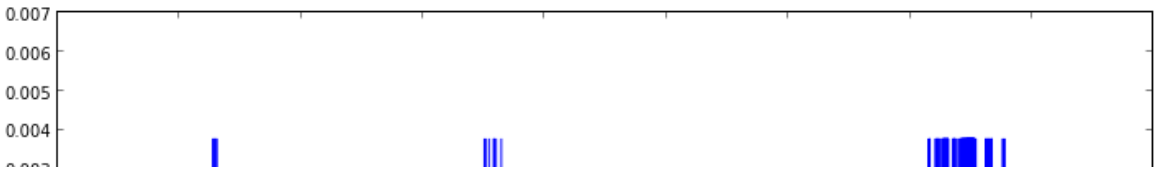
In [174]:
```
step_sense('porta')
show_filter()
```

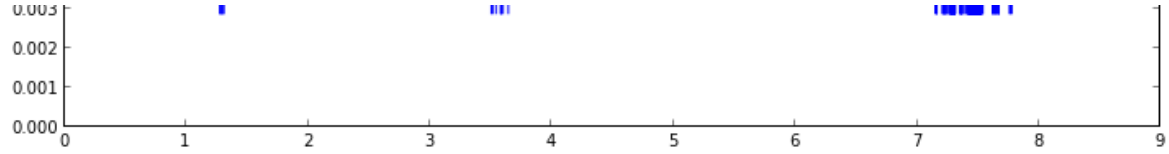


In [175]:
```
step_resample()
show_filter()
```

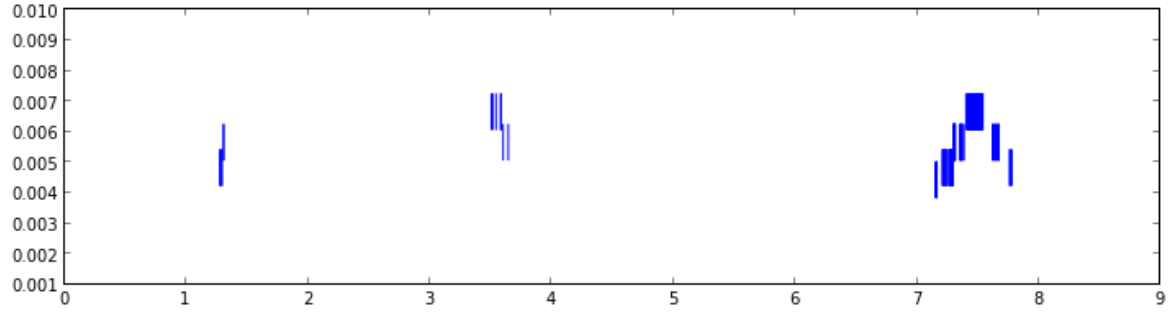


In [176]:
```
step_move(3.8)
show_filter()
```

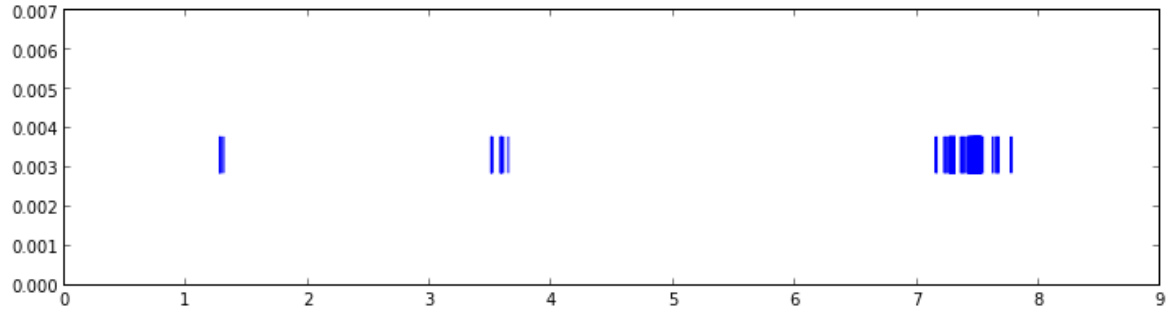




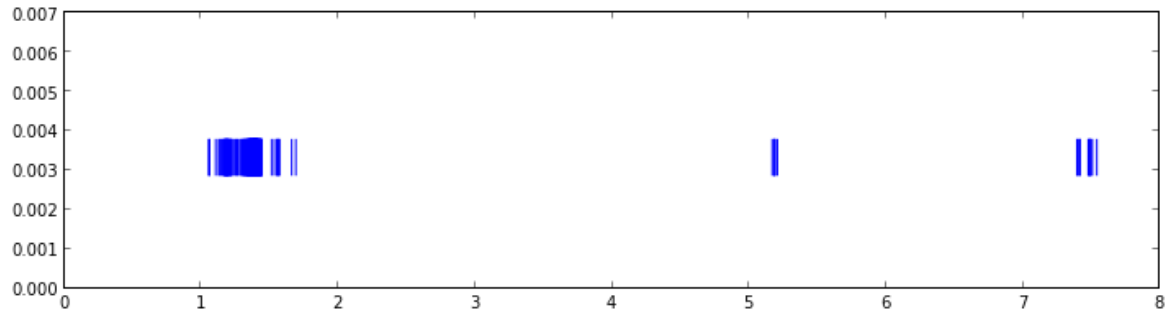
In [177]:
```
step_sense('porta')
show_filter()
```



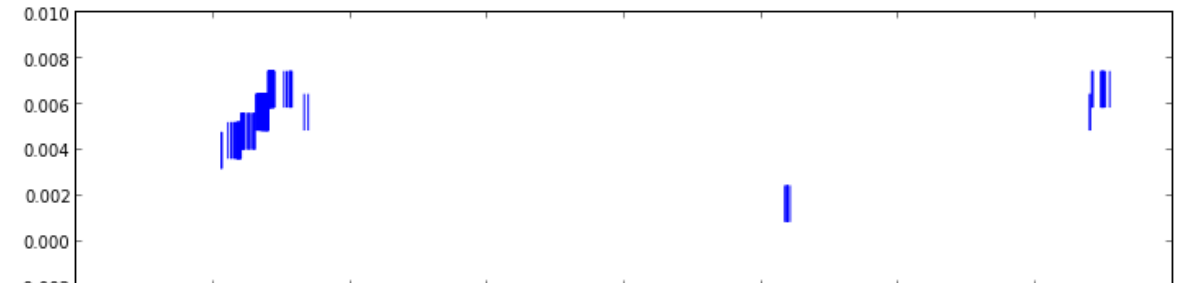
In [178]:
```
step_resample()
show_filter()
```



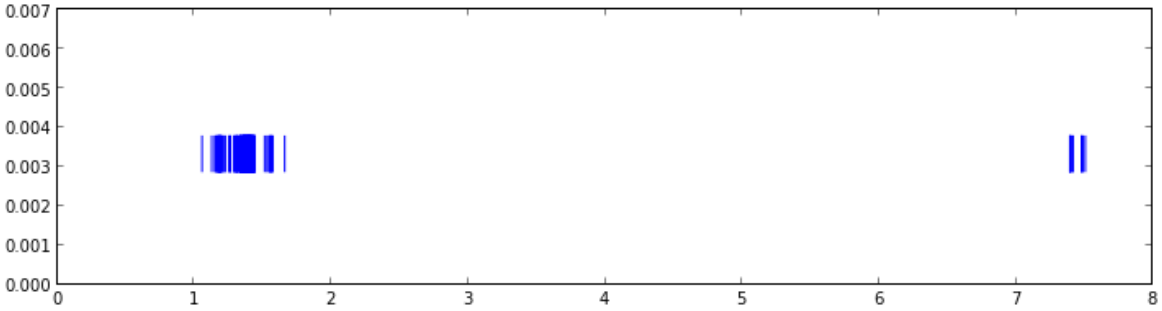
In [179]:
```
step_move(3.9)
show_filter()
```



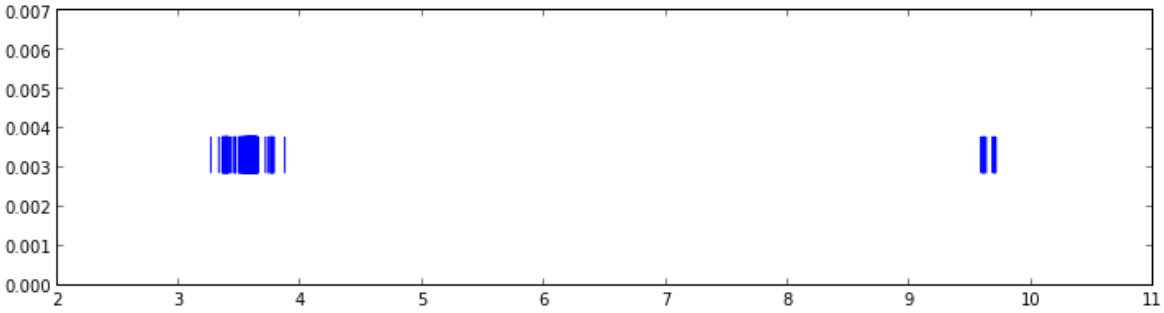
In [180]:
```
step_sense('porta')
show_filter()
```



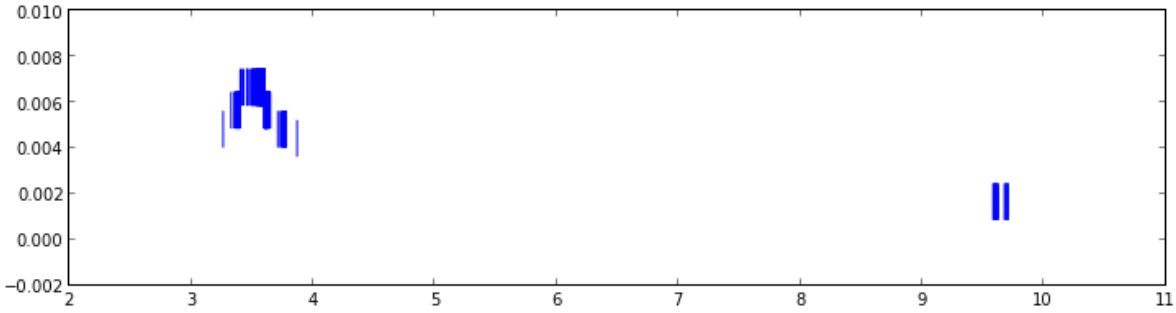
In [181]:
```
step_resample()
show_filter()
```



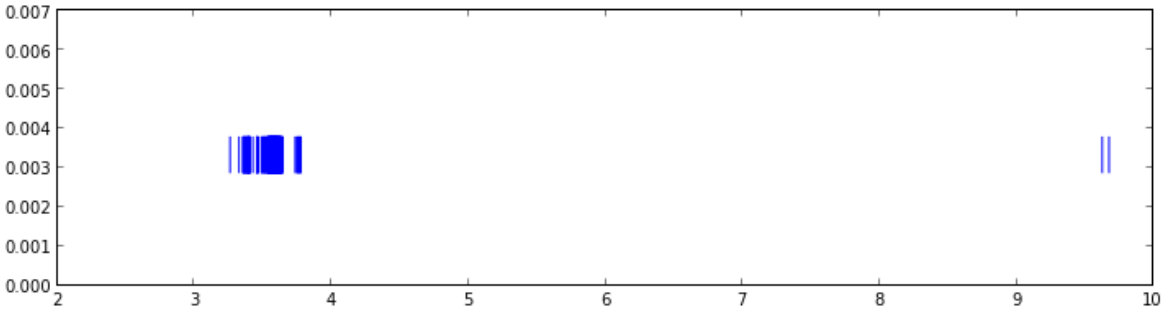
In [182]:
```
step_move(2.2)
show_filter()
```



In [183]:
```
step_sense('porta')
show_filter()
```

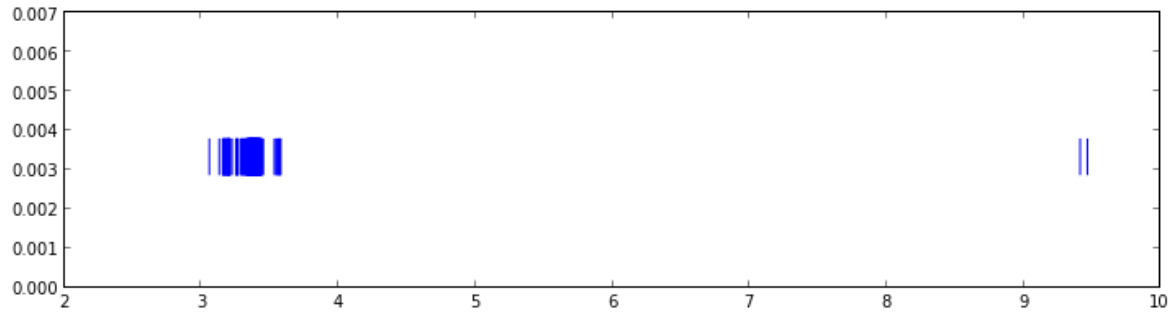


In [184]:
```
step_resample()
show_filter()
```

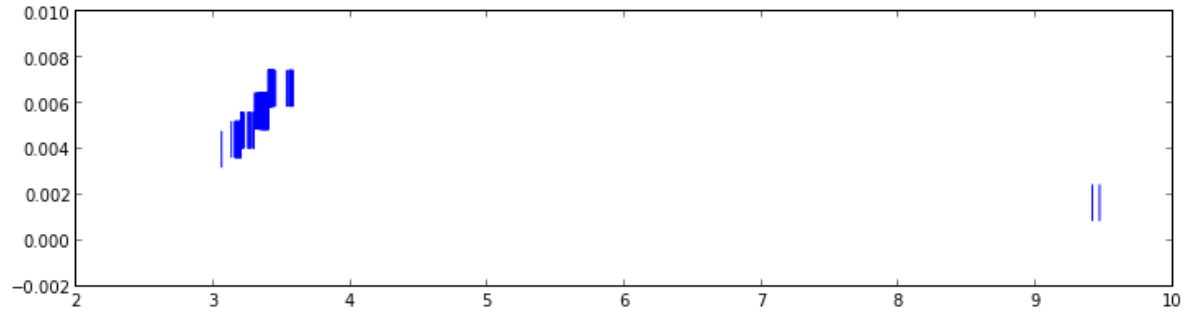


In [185]:
```
step_move(-0.2)
```

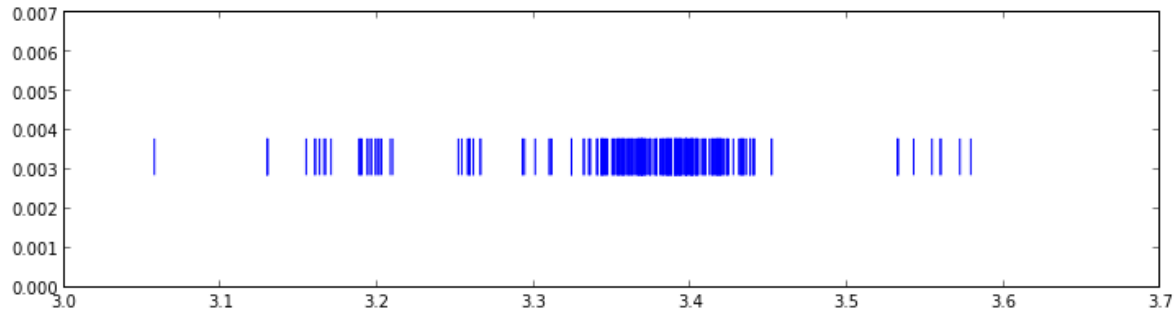
```
show_filter()
```



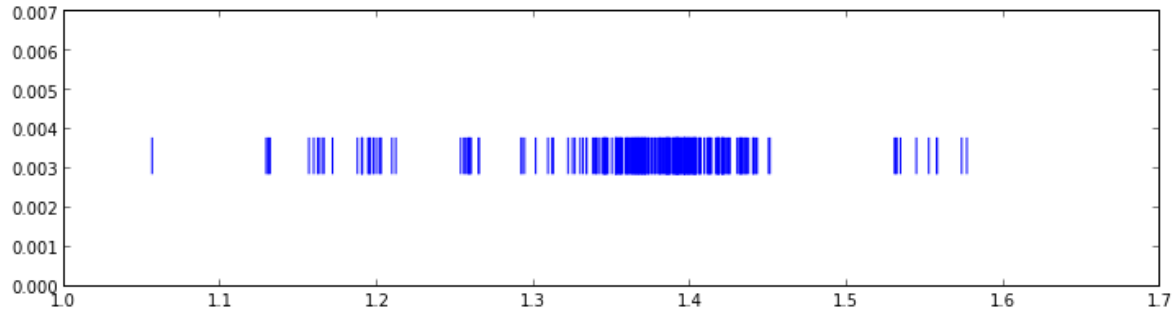
In [186]:
```
step_sense('porta')
show_filter()
```



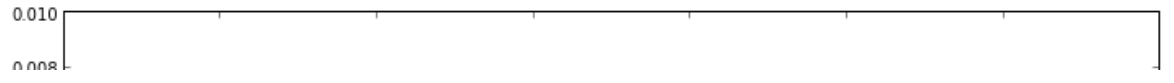
In [187]:
```
step_resample()
show_filter()
```

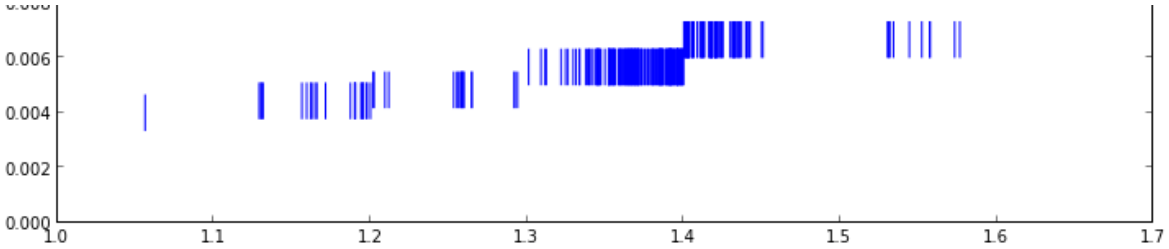


In [188]:
```
step_move(-2)
show_filter()
```

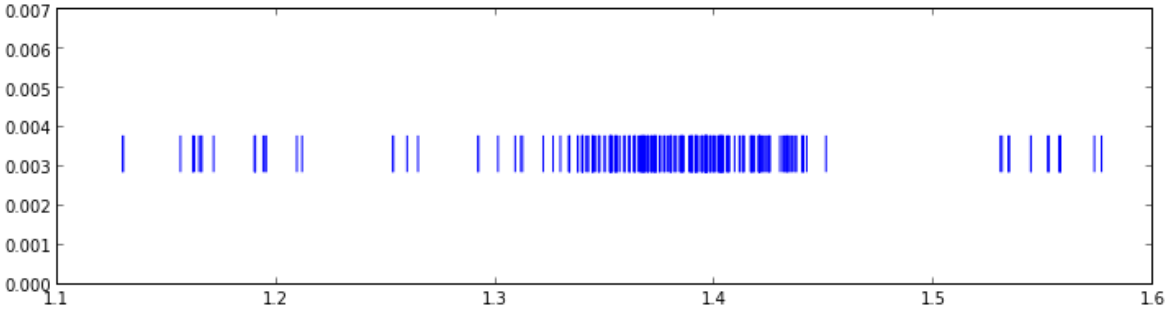


In [189]:
```
step_sense('porta')
show_filter()
```

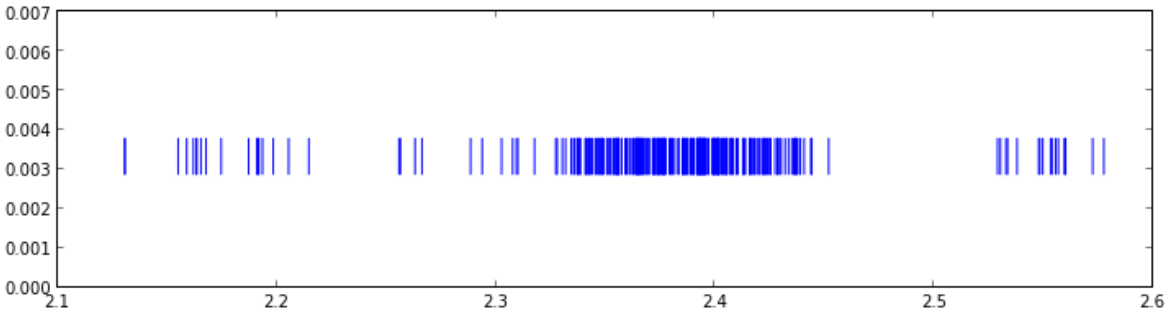




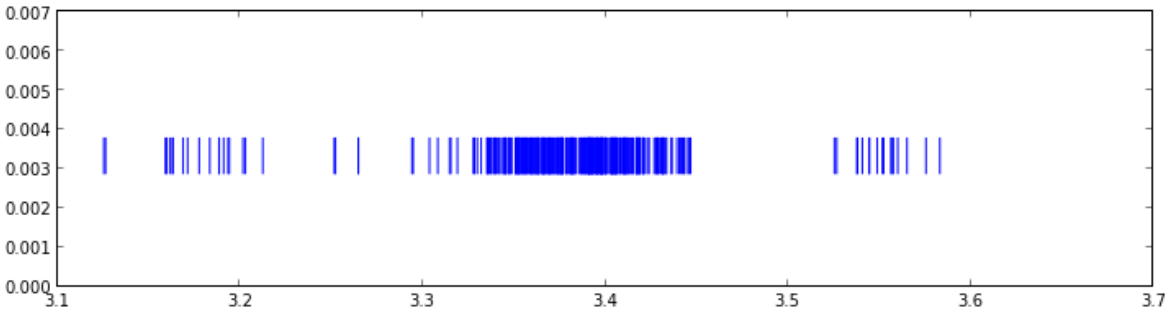
In [190]:
```
step_resample()
show_filter()
```



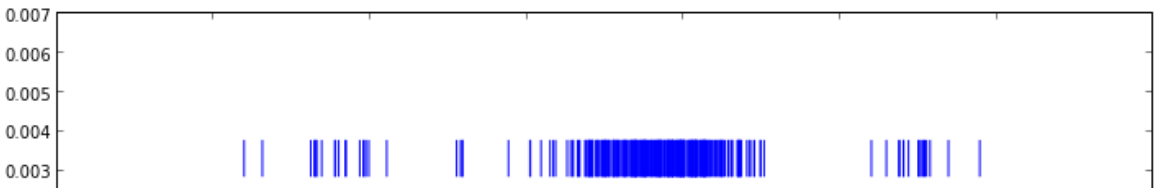
In [191]:
```
step_move(1)
show_filter()
```

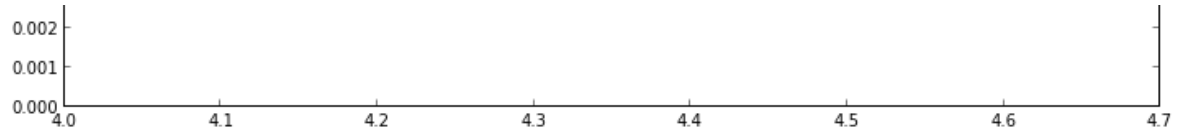


In [192]:
```
step_move(1)
show_filter()
```

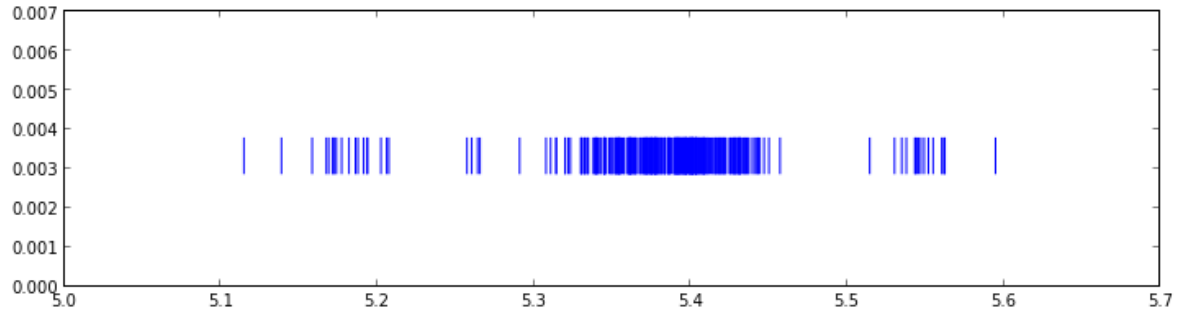


In [193]:
```
step_move(1)
show_filter()
```

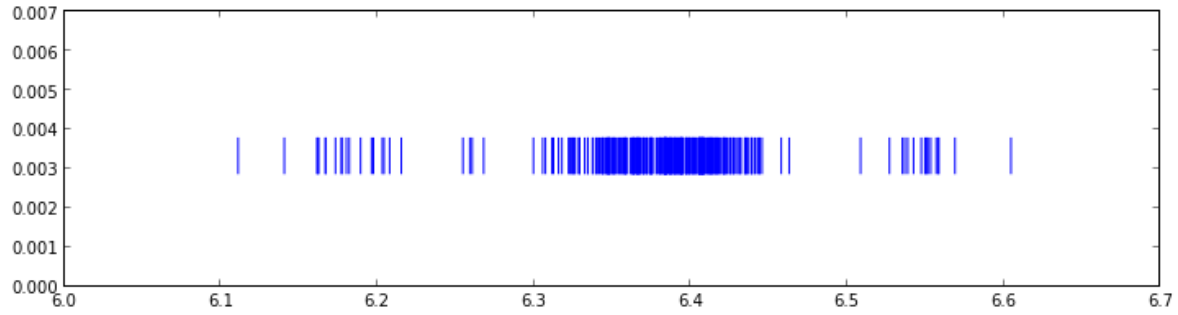




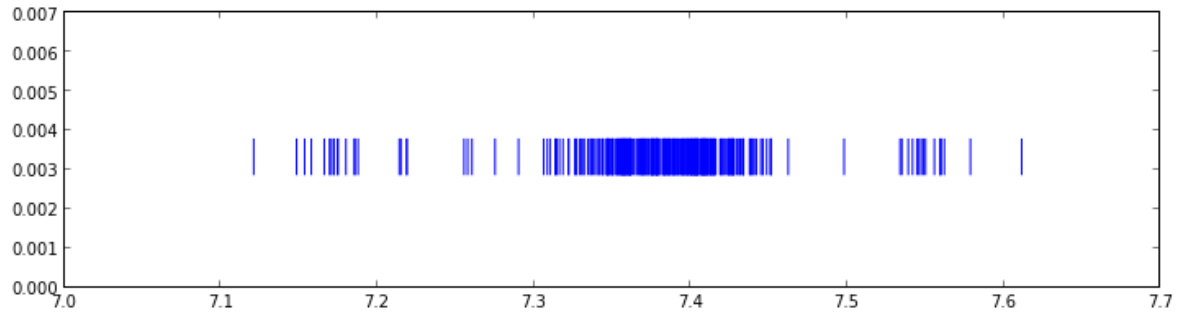
In [194]:
```
step_move(1)
show_filter()
```



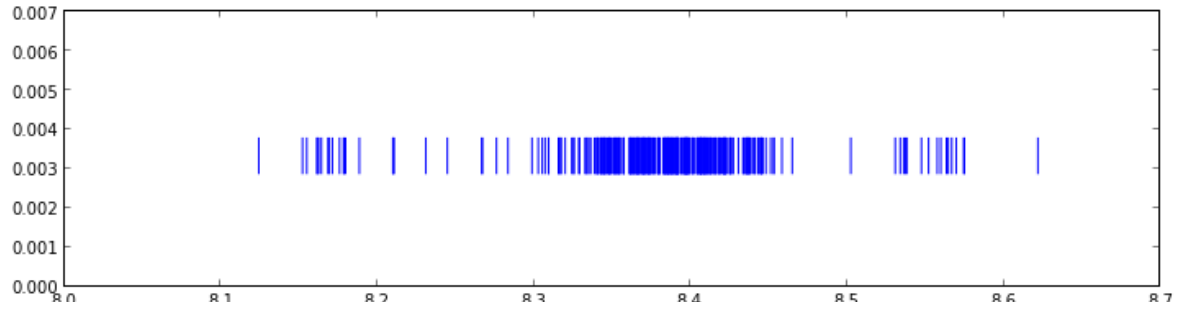
In [195]:
```
step_move(1)
show_filter()
```



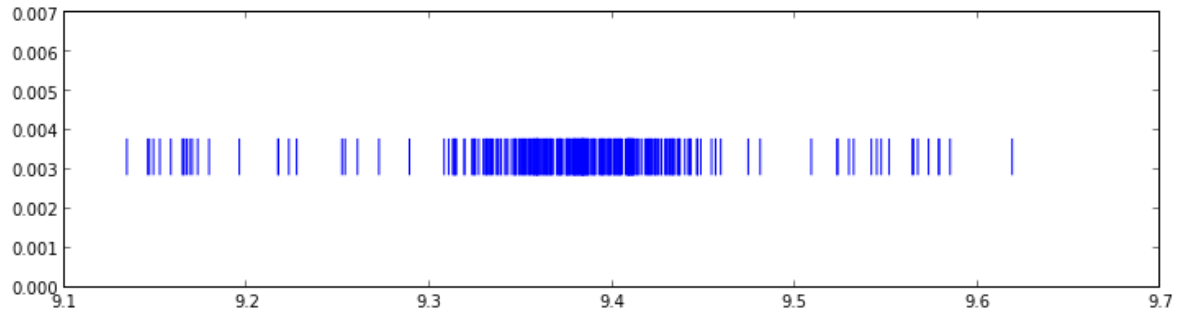
In [196]:
```
step_move(1)
show_filter()
```



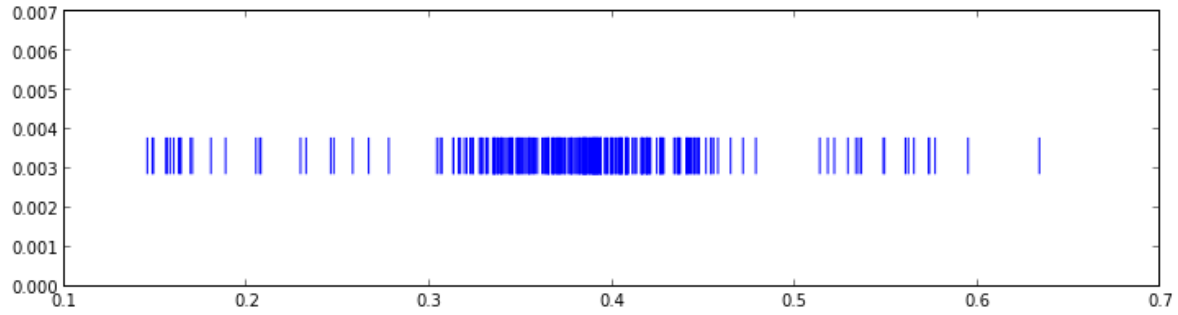
In [197]:
```
step_move(1)
show_filter()
```



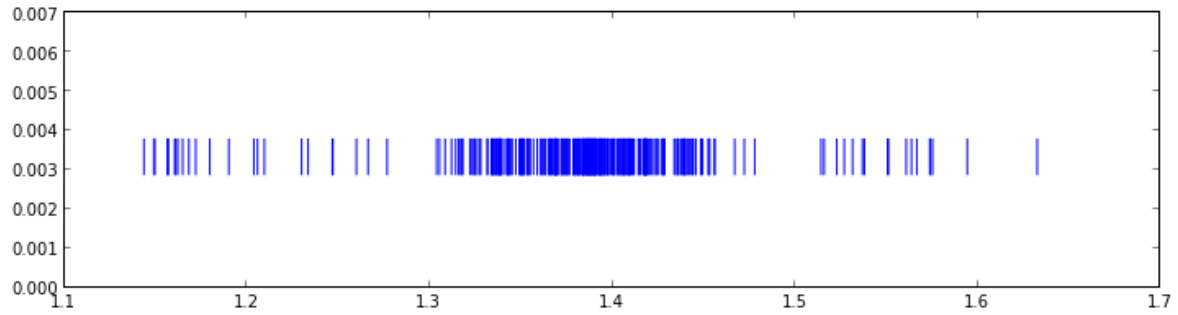
In [198]:
```
step_move(1)
show_filter()
```



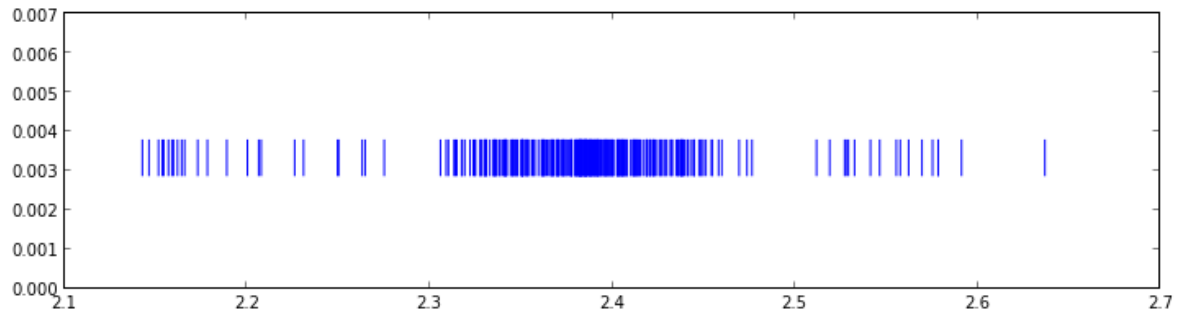
In [199]:
```
step_move(1)
show_filter()
```



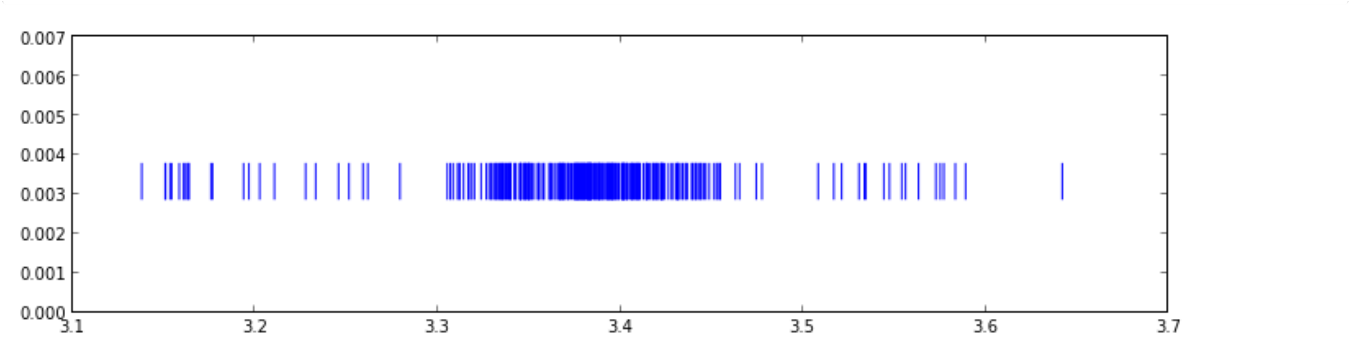
In [200]:
```
step_move(1)
show_filter()
```



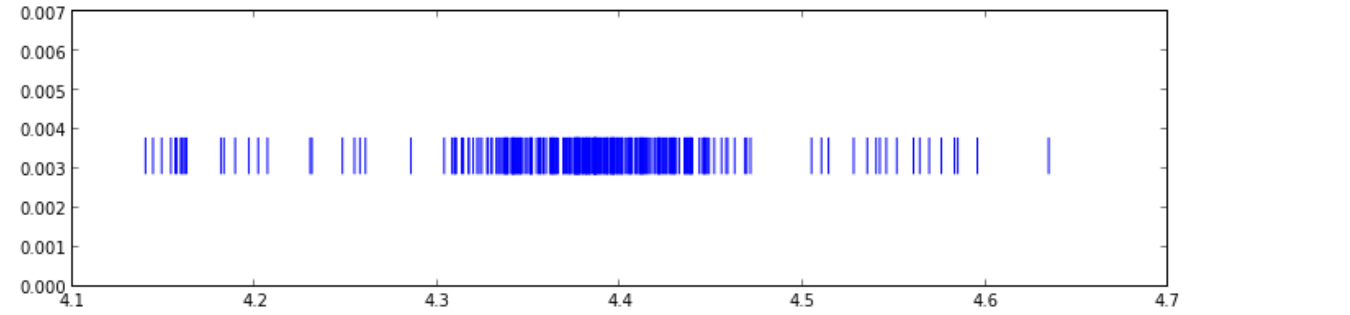
In [201]:
```
step_move(1)
show_filter()
```



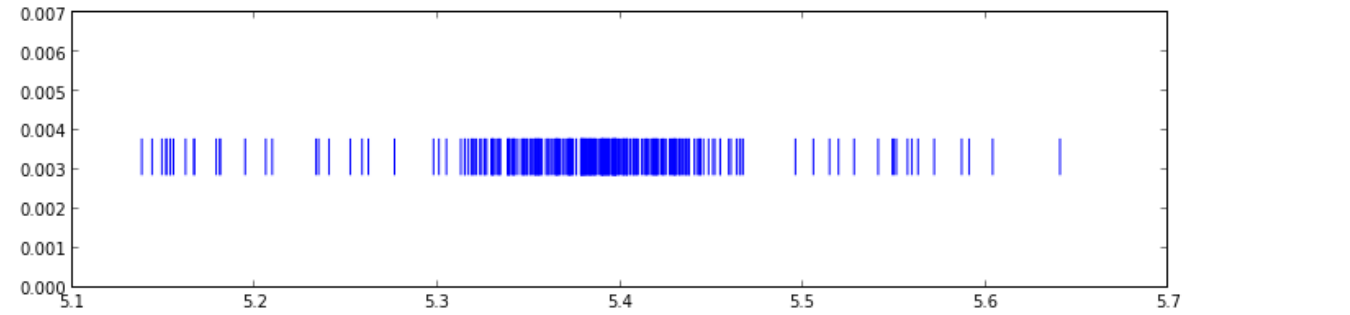
In [202]:
```
step_move(1)
show_filter()
```



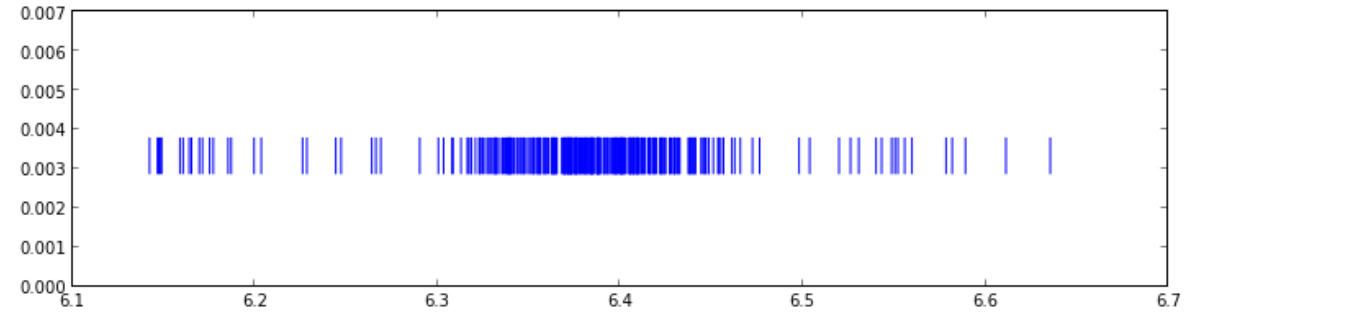
In [203]:
```
step_move(1)
show_filter()
```



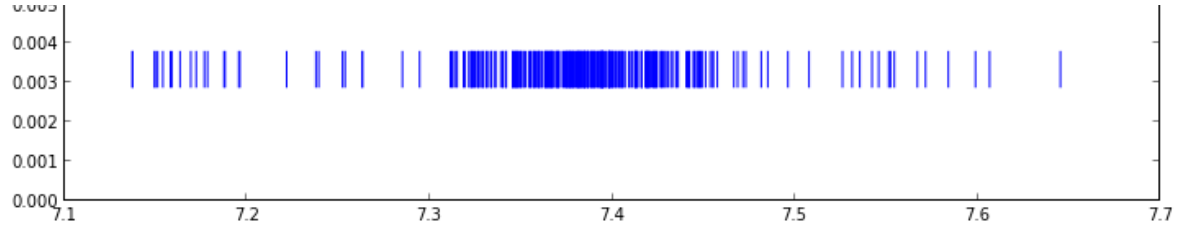
In [204]:
```
step_move(1)
show_filter()
```



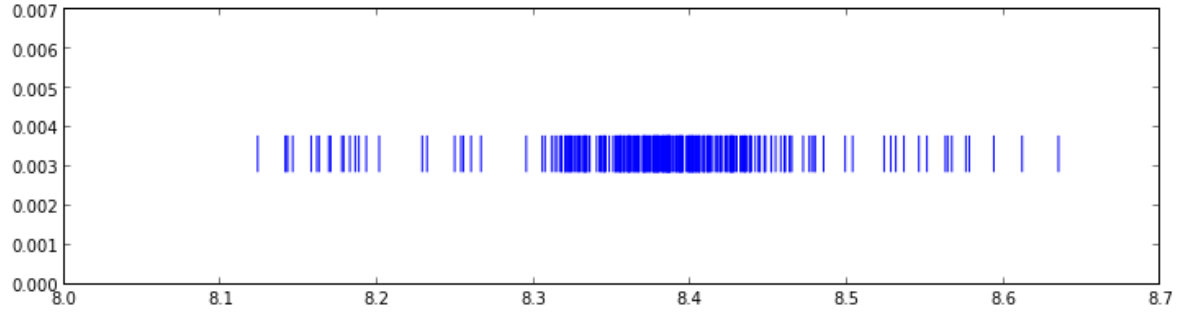
In [205]:
```
step_move(1)
show_filter()
```



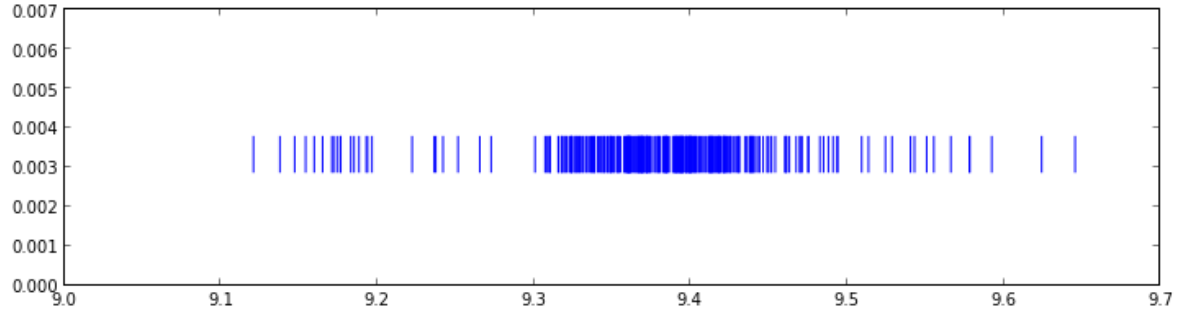
In [206]:
```
step_move(1)
show_filter()
```



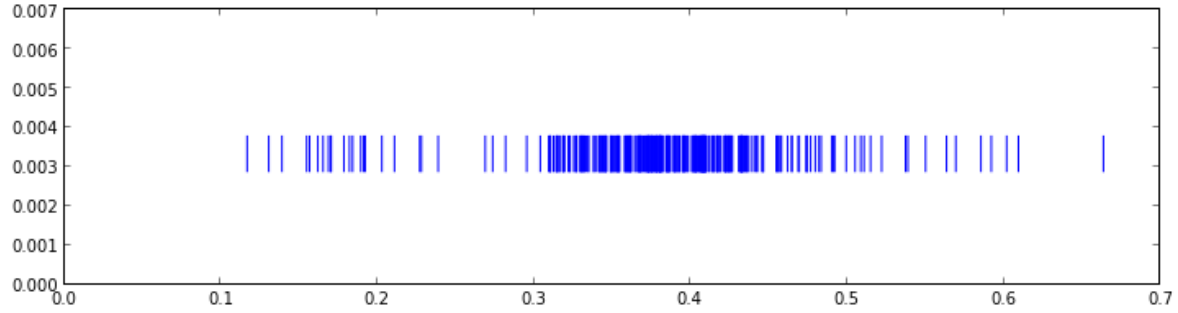
In [207]:
```
step_move(1)
show_filter()
```



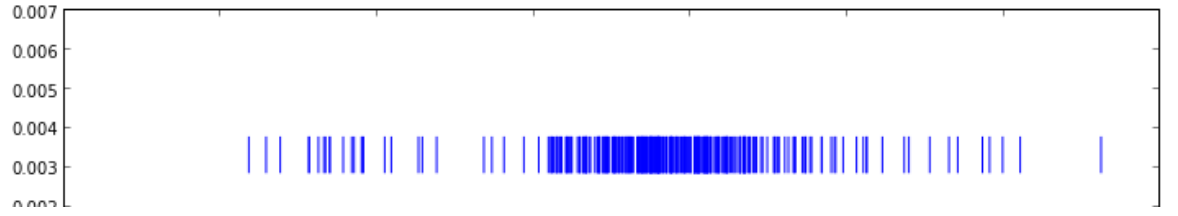
In [208]:
```
step_move(1)
show_filter()
```



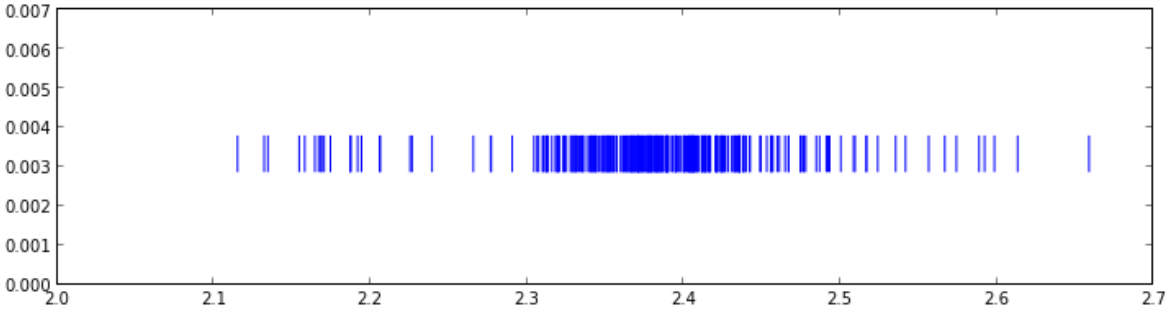
In [209]:
```
step_move(1)
show_filter()
```



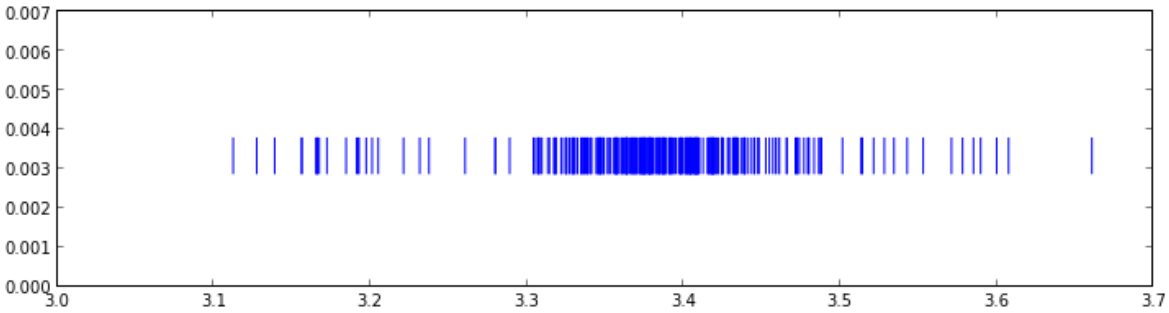
In [210]:
```
step_move(1)
show_filter()
```



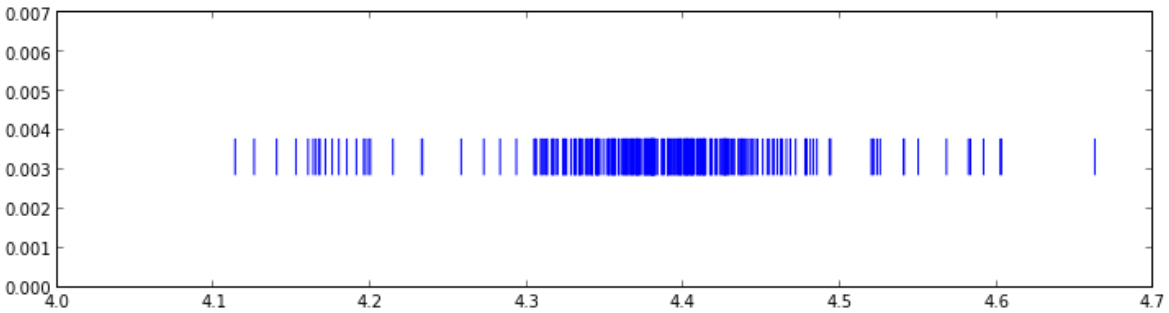
In [211]:
```
step_move(1)
show_filter()
```



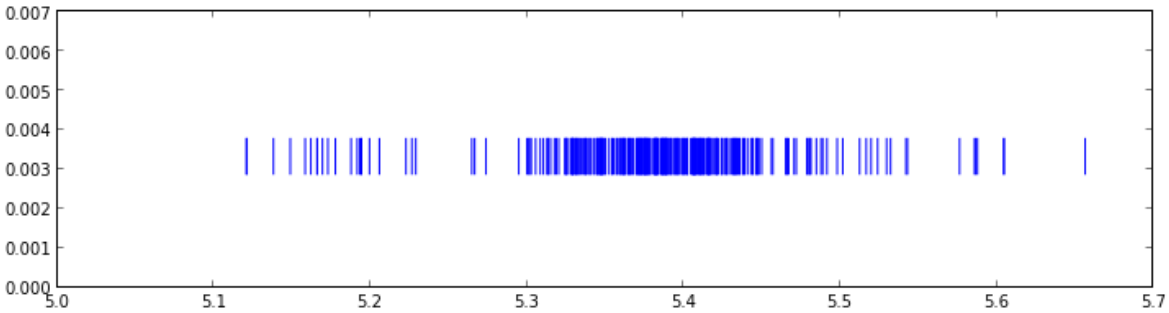
In [212]:
```
step_move(1)
show_filter()
```



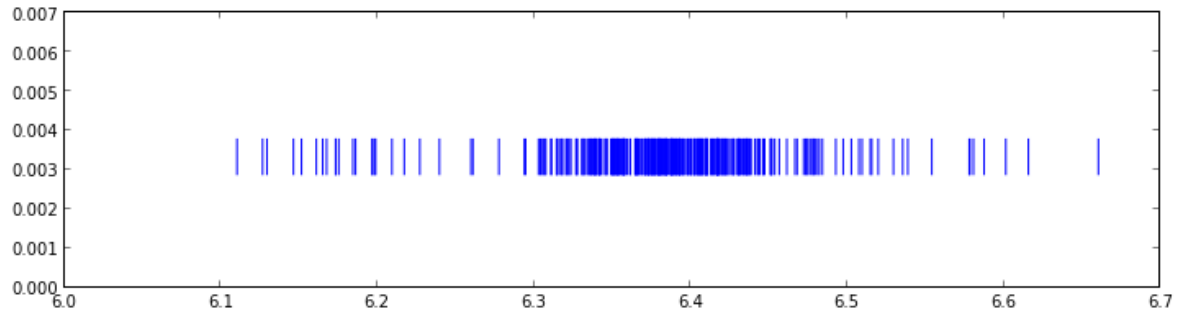
In [213]:
```
step_move(1)
show_filter()
```



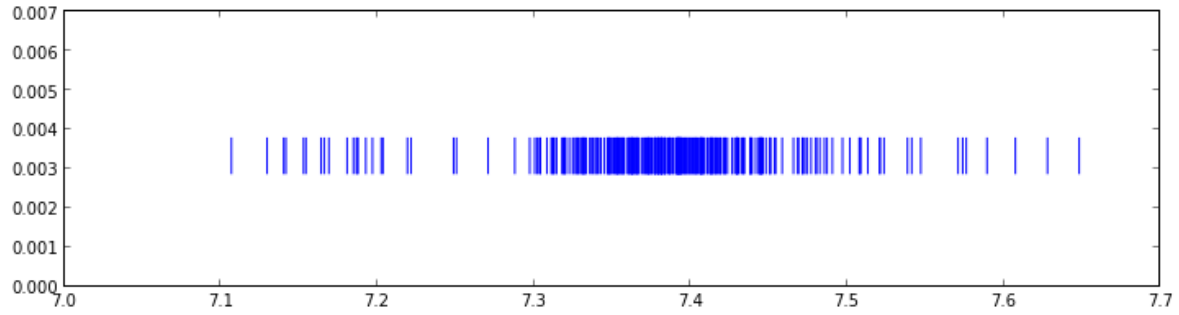
In [214]:
```
step_move(1)
show_filter()
```



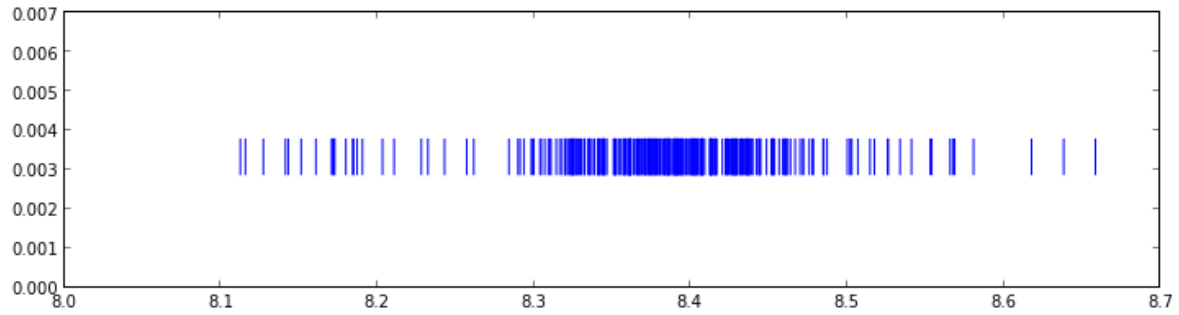
In [215]:
```
step_move(1)
show_filter()
```



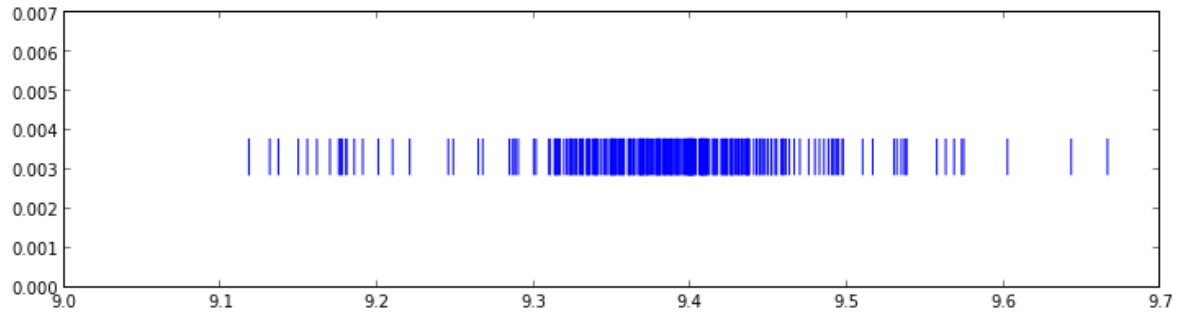
In [216]:
```
step_move(1)
show_filter()
```



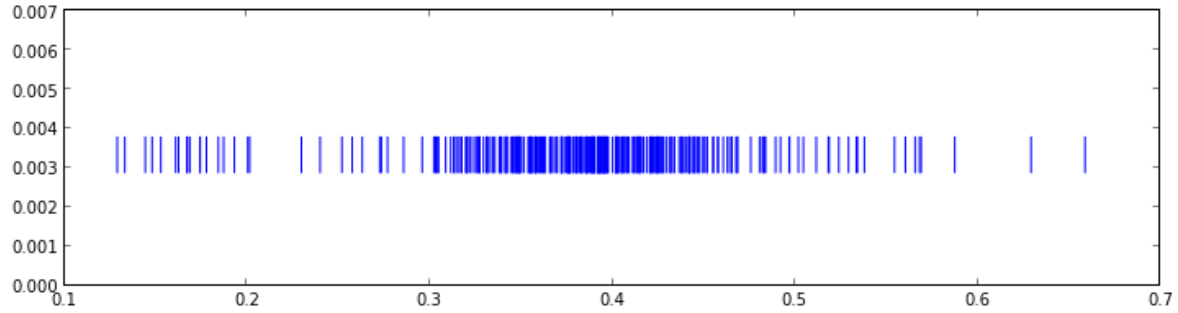
In [217]:
```
step_move(1)
show_filter()
```



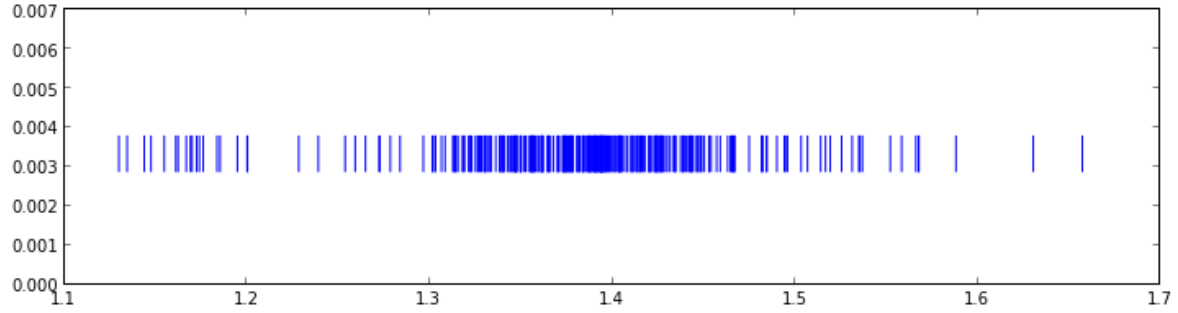
In [218]:
```
step_move(1)
show_filter()
```



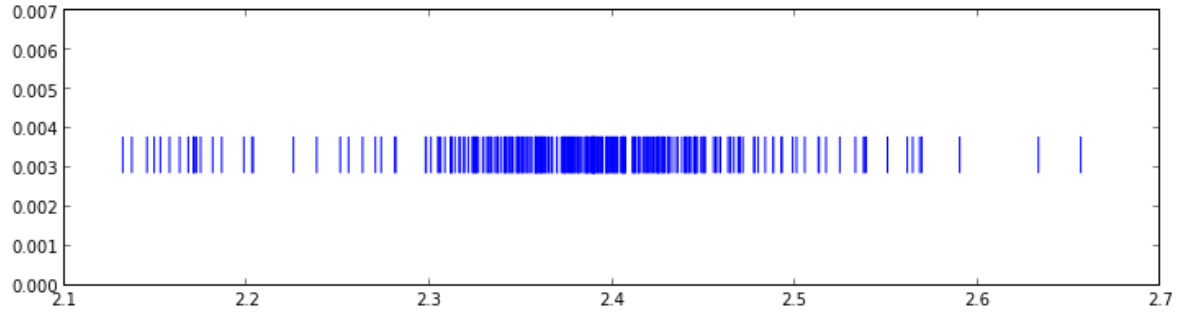
In [219]:
```
step_move(1)
show_filter()
```



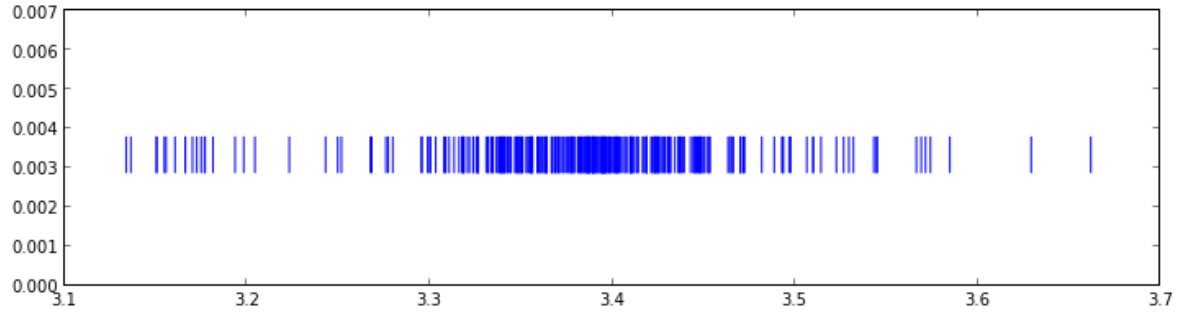
In [220]:
```
step_move(1)
show_filter()
```



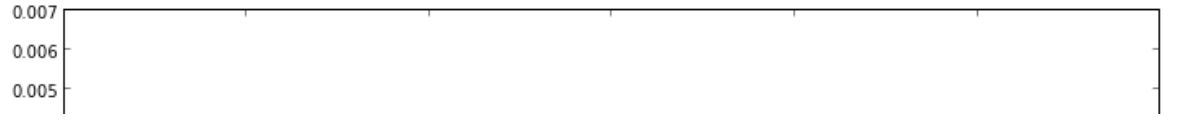
In [221]:
```
step_move(1)
show_filter()
```

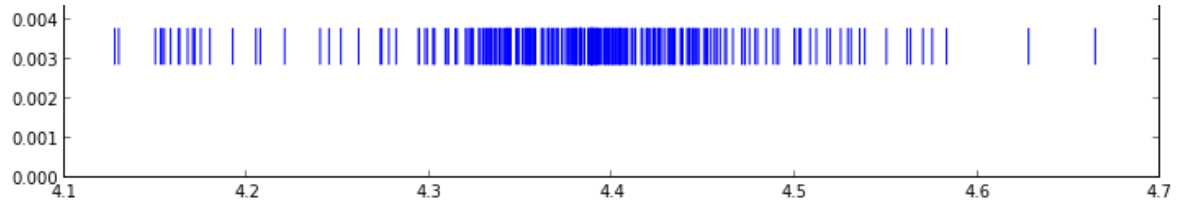


In [222]:
```
step_move(1)
show_filter()
```

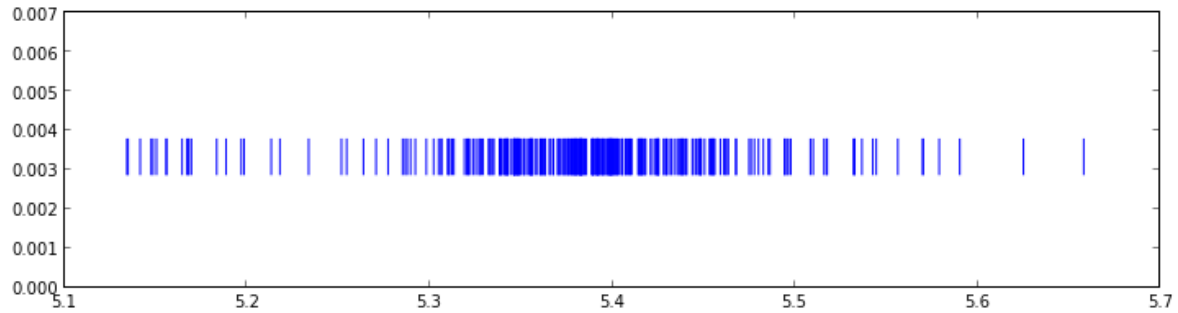


In [223]:
```
step_move(1)
show_filter()
```

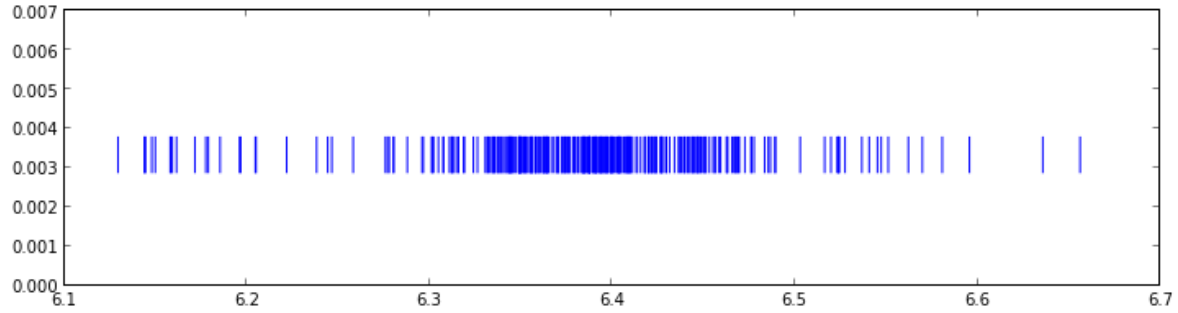




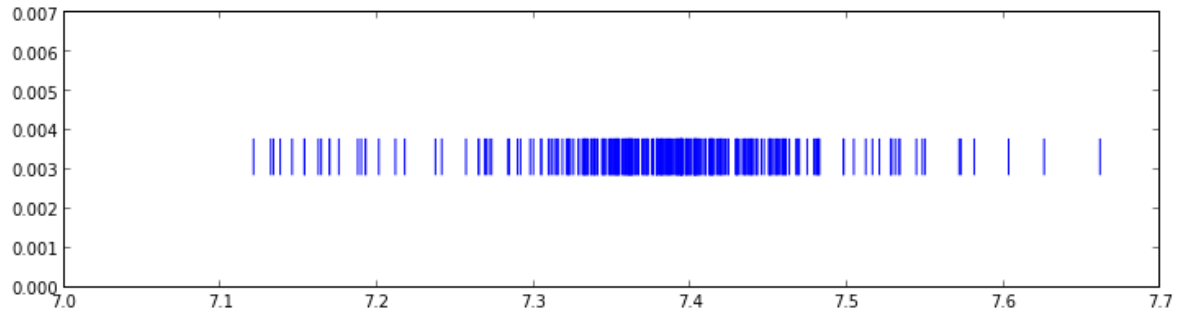
In [224]:
```
step_move(1)
show_filter()
```



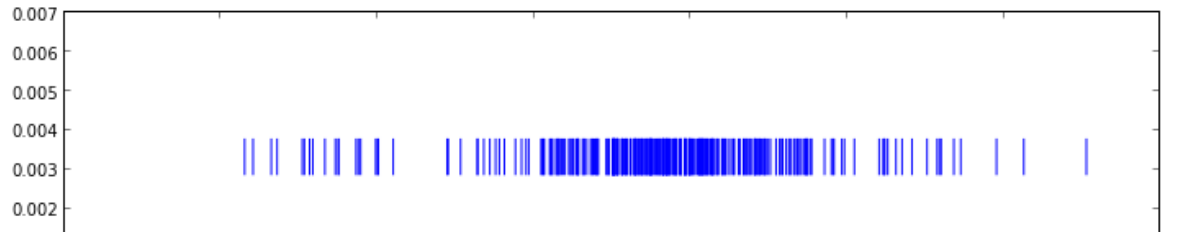
In [225]:
```
step_move(1)
show_filter()
```



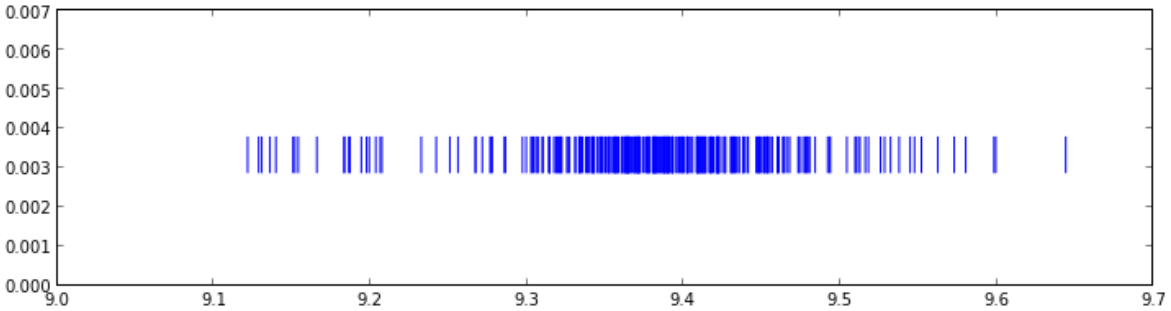
In [226]:
```
step_move(1)
show_filter()
```



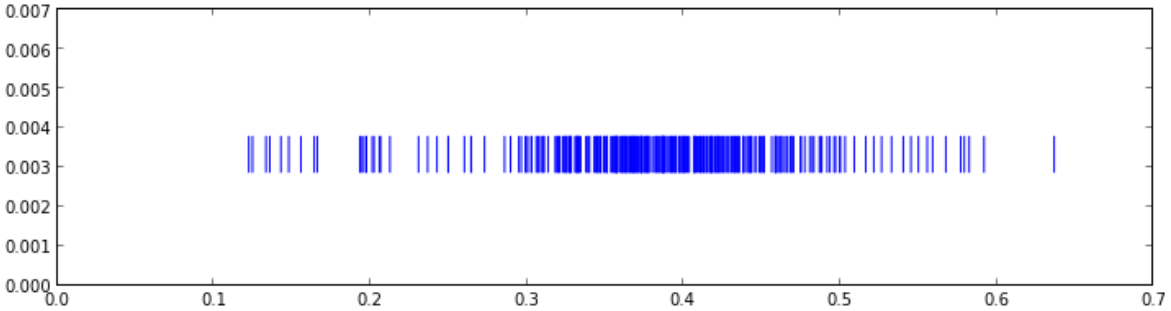
In [227]:
```
step_move(1)
show_filter()
```



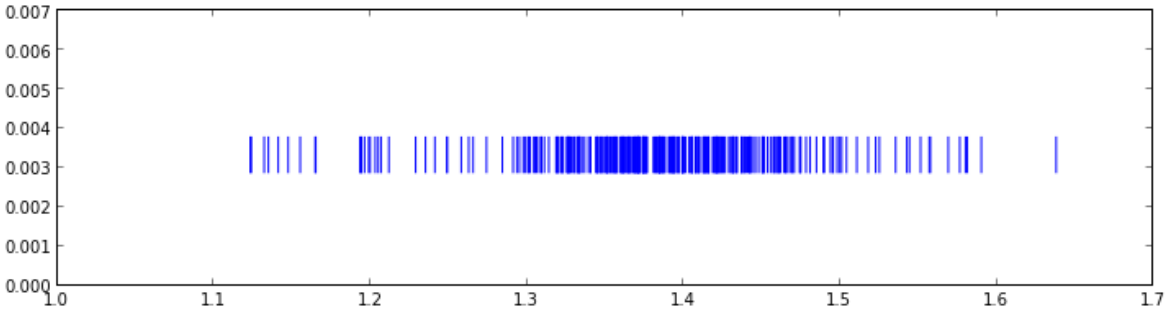
In [228]:
```
step_move(1)
show_filter()
```



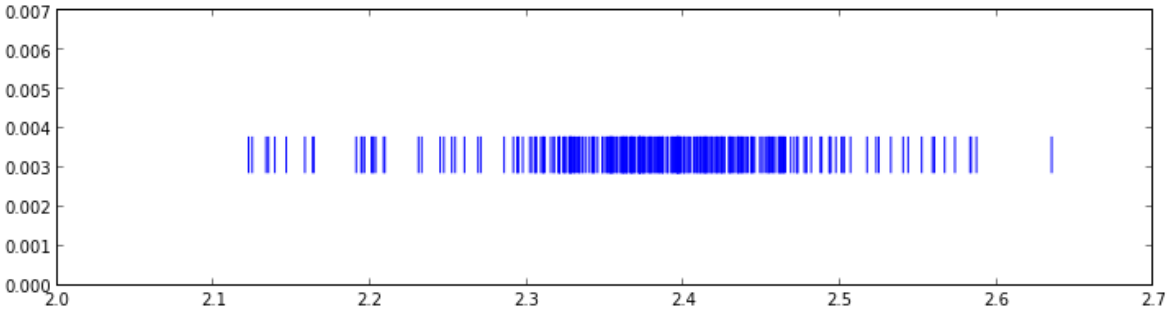
In [229]:
```
step_move(1)
show_filter()
```



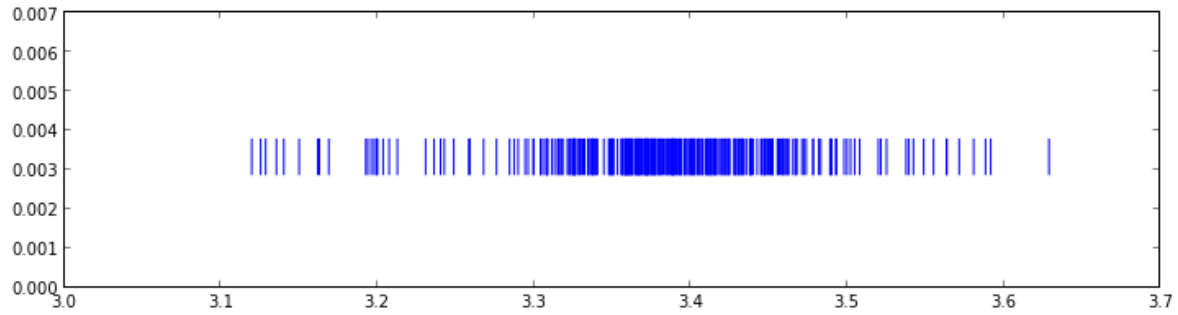
In [230]:
```
step_move(1)
show_filter()
```



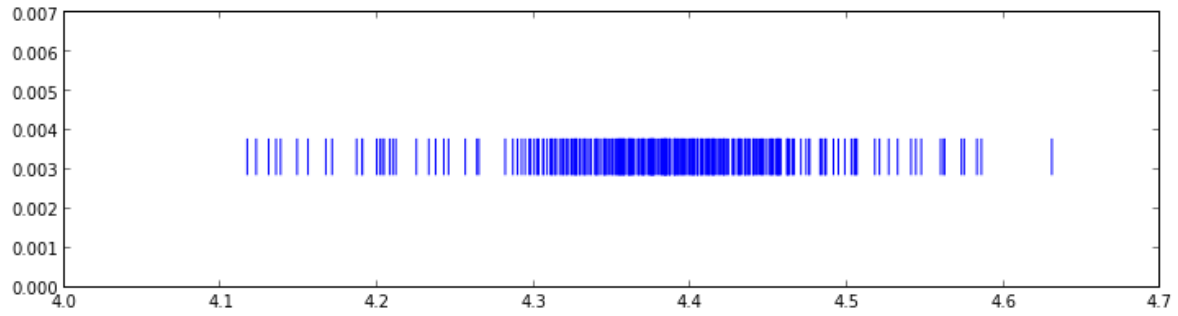
In [231]:
```
step_move(1)
show_filter()
```



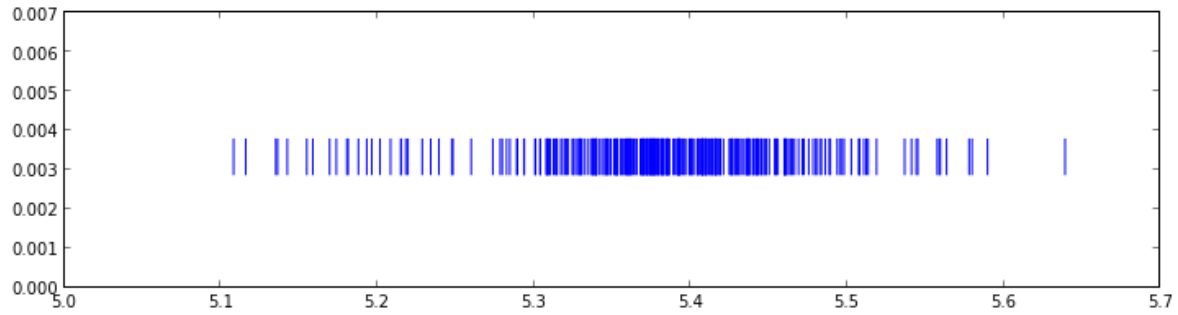
In [232]:
```
step_move(1)
show_filter()
```



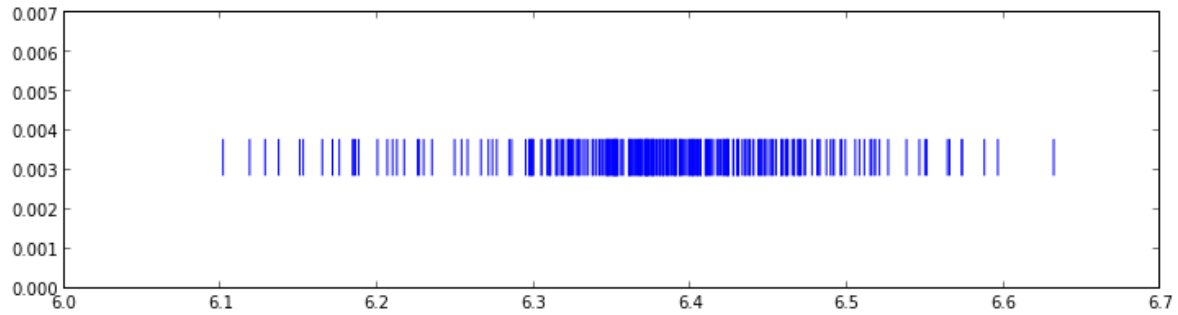
In [233]:
```
step_move(1)
show_filter()
```



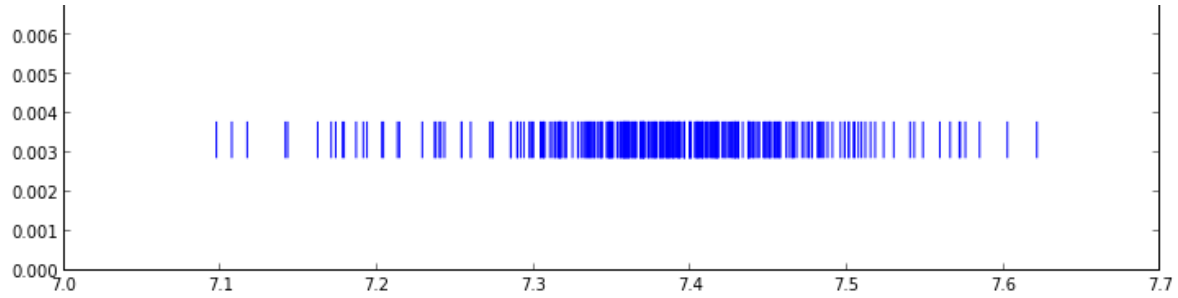
In [234]:
```
step_move(1)
show_filter()
```



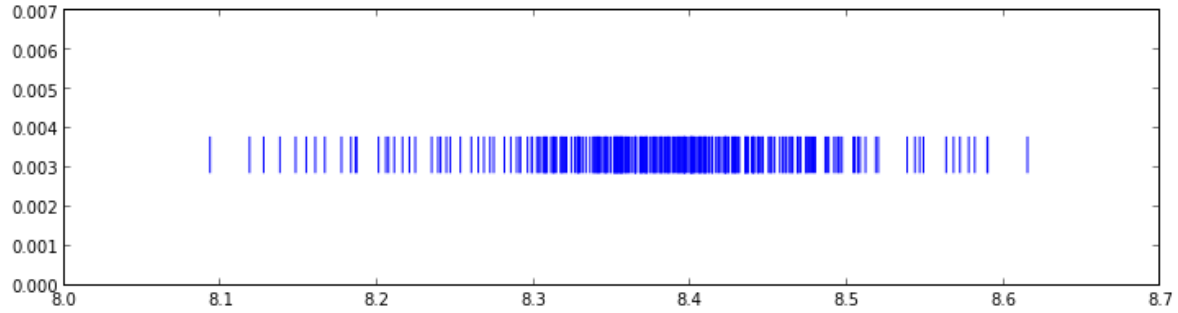
In [235]:
```
step_move(1)
show_filter()
```



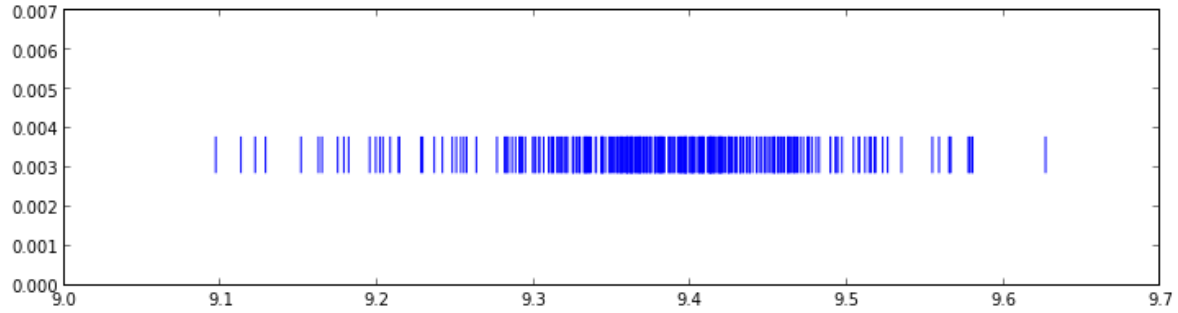
In [236]:
```
step_move(1)
show_filter()
```



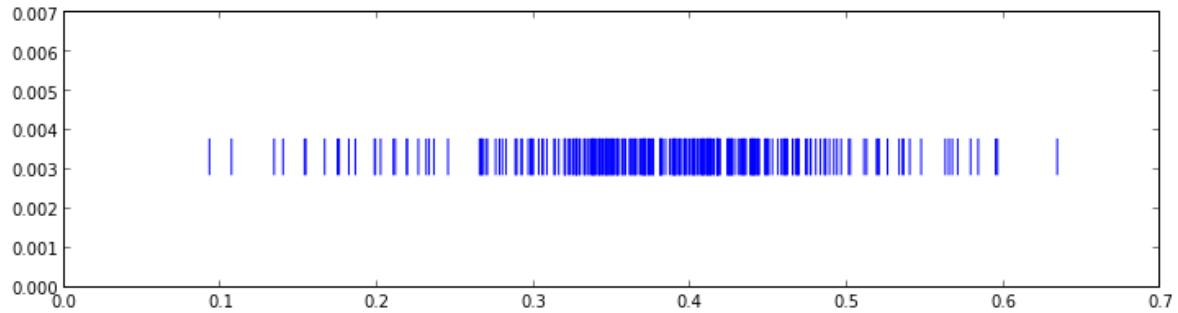
In [237]:
```
step_move(1)
show_filter()
```



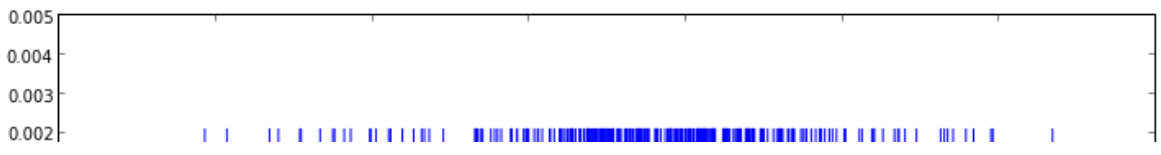
In [238]:
```
step_move(1)
show_filter()
```

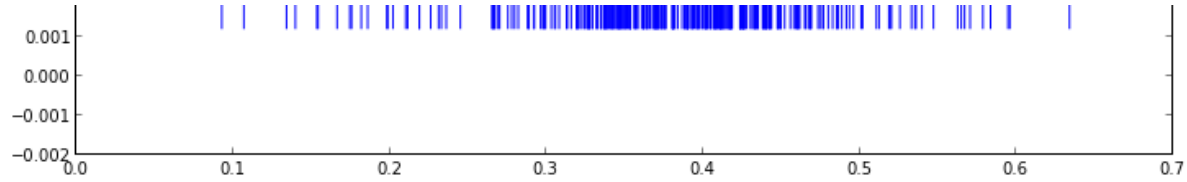


In [239]:
```
step_move(1)
show_filter()
```

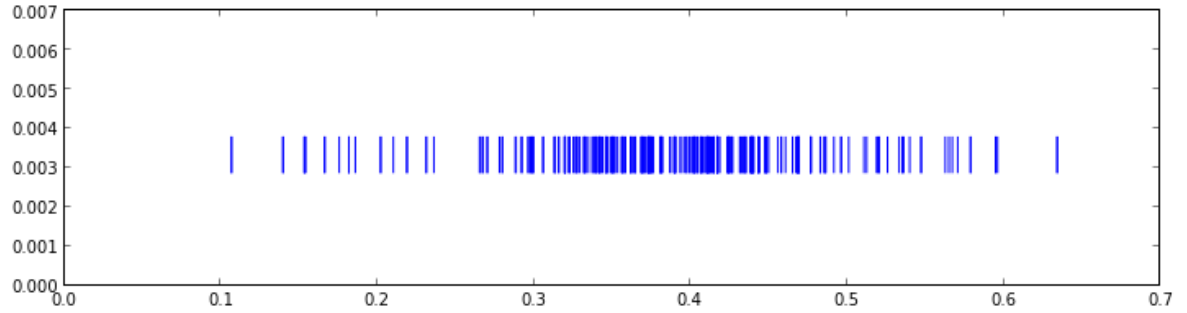


In [240]:
```
step_sense('porta')
show_filter()
```





In [241]:
```
step_resample()
show_filter()
```



In [ ]: