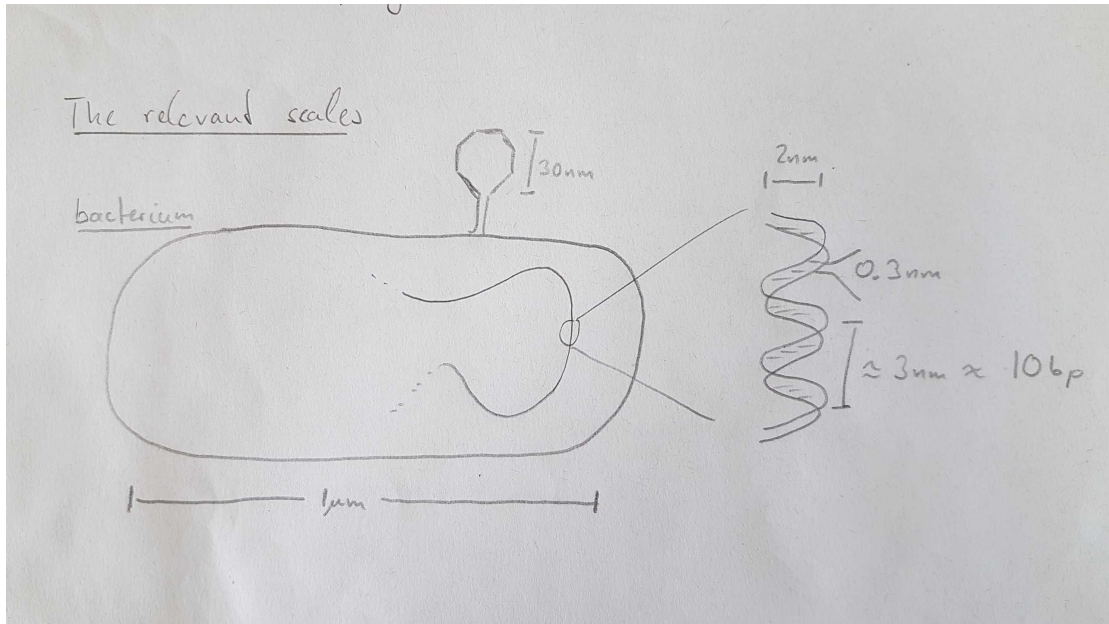
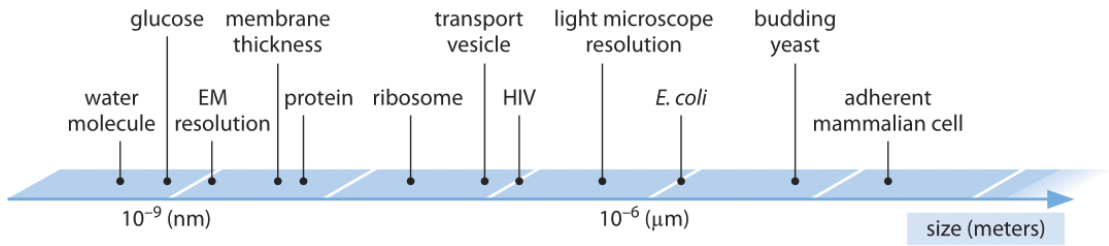


# Relevant linear dimensions



Life operates on all scales from the sub-nanometer molecules to organisms. The range of these length scales is well illustrated in this graph:



source: Physical Biology of the Cell

Understanding the relevant length scales is crucial for understanding many biological processes and getting a sense of why things are fast, slow etc.

```
In [1]: # Note that these numbers are rough estimates and often are only represen
# values picked from a range. They are intended to give you a sense of th
# relative sizes but are not an authoritative source.
linear_dimensions = { # in meters
    "H2O": 2.45e-10,
    "dsDNA base pair height": 3.4e-10,
    "glucose": 9e-10,
    "dsDNA diameter": 2e-9,
    "membrane thickness": 3e-9,
    "ribosome": 2.5e-8,
    "RNA virus": 1e-7,
    "wave length of visible light": 5e-7,
    "bacterium": 1e-6,
    "budding yeast": 5e-6,
    "mammalian cell": 1e-5,
    "human oocyte": 1e-4,
    "fruit fly egg": 5e-4,
    "chicken egg": 5e-2,
    "human": 1.8,
    "blue whale": 25
}

print(f"size of water molecules: {linear_dimensions['H2O']}m")
print(f"size of RNA virus: {linear_dimensions['RNA virus']}m")
```

```
size of water molecules: 2.45e-10m
size of RNA virus: 1e-07m
```

The data structure above is called a "dictionary" as it associates values (numbers measured in meters) to keys. These values can be accessed as

```
linear_dimensions['H2O'] .
```

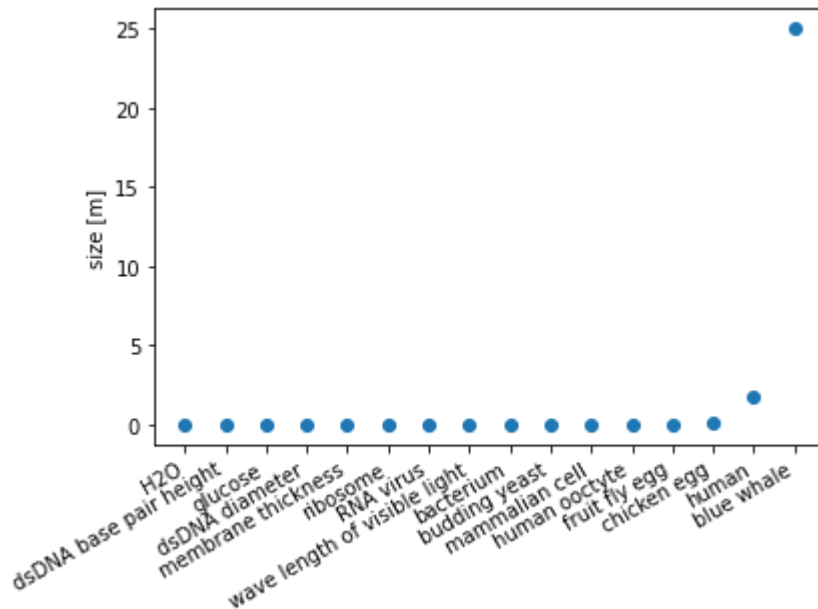
Let's graph these numbers. We can do this with the plotting library `matplotlib`. After the import, the plotting commands are available under the name `plt`.

```
In [2]: import matplotlib.pyplot as plt
```

```
In [3]: names = list(linear_dimensions.keys()) # make a list of the names
sizes = [linear_dimensions[k] for k in names] # make a list of the value
print(names)
plt.plot(sizes, 'o') # the 'o' here indicates that we use a circle marke
plt.xticks(range(len(names)), names, rotation=30, ha='right') # label the
plt.ylabel('size [m]') # label the y-axis
```

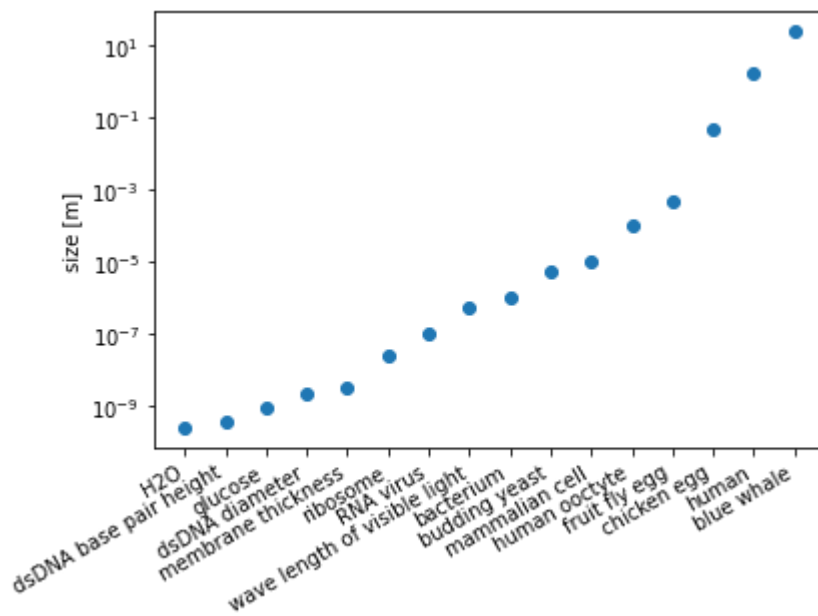
```
['H2O', 'dsDNA base pair height', 'glucose', 'dsDNA diameter', 'membrane t
hickness', 'ribosome', 'RNA virus', 'wave length of visible light', 'bacte
rium', 'budding yeast', 'mammalian cell', 'human oocyte', 'fruit fly egg
', 'chicken egg', 'human', 'blue whale']
```

```
Out[3]: Text(0, 0.5, 'size [m]')
```



As you see, we don't see much. Only for whale and the human it is visible that the values are non-zero. This is not surprising, we are plotting quantities on vastly different scales. To get a better sense of the relative magnitude of points in our data, it is useful to look at these data on a logarithmic scale:

```
In [4]: plt.plot(sizes, 'o')
plt.xticks(range(len(names)), names, rotation=30, ha='right')
plt.yscale('log')
plt.ylabel('size [m]')
plt.show()
```



Now the differences between all data points are clearly visible. One a logarithmic scale, fold changes are additive. Thus it 10 is as far from 1 and 100 is from 10. This fact is a consequence of the fundamental property of logarithms:

$$\log(xy) = \log(x) + \log(y)$$

## Additional resources

- video lesson on logarithmic scales by the [Khan academy](#)
- [Biology by the Numbers](#) by Milo and Phillips, Chapter 1

In [ ]: