UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Corso di Laurea in Fisica

Tesi di Laurea

Charge carrier mobilities in Metal Halide Perovskites, basic overview and calculation through GW simulations

Relatore	Laureando
Prof. Paolo Umari	Lorenzo Barbiero

Anno Accademico 2021/2022

Abstract

Metal Halide Perovskites are a class of materials with very promising photo-electronic capabilities that, coupled with easy manufacturing, makes them a suitable candidate for the solar cells of the future.
In this thesis work the goal is to evaluate electron and hole mobilities for such compounds starting from first principles GW simulations, focusing especially on vibrational modes for infrared scattering in order to better understand different frequencies and atoms contribution to mobility and to develop a framework for the study of future compounds.

Contents

Introduction

Metal Halide Perovskites (MHPs) are hybrid organic-inorganic semiconductors which have shown amazing capabilities in photovoltaic applications and solar cells production. It has indeed been shown that MHP based solar cells can reach efficiencies on par with traditional silicon based ones, moreover production through chemical reactions makes them exceptionally cheaper to produce.

The crystal stucture is typical of the "perovskite" class of materials, in which a metallic atom is surrounded by an octahedra of halides, with cations sitting in the interstices. This structure opens up a lot of possibilities for new compounds since the three main elements are independent from one another and, thus, can be freely modified.

One of the main parameters which dictates the behaviour of a solar cell is the electron/hole mobility of the employed semiconductors, the objective of this thesis work is a first principle study of such mobilities for a various set of metal halide perovskites utilising a new method developed by prof. Paolo Umari et. al.

Despite being based on multi-body GW perturbations smart optimisations lead to the possibility of performing these calculations on standard personal computers in reasonable times.
After a brief introduction on the basic theoretical principles of semiconductor mobility and GW simulations an analysis on infrared vibrations for phonon scattering is performed, especially focusing on the participation of the various atomic species to vibrations in order to highlight potential synergies in the compounds which will need to be addressed in future researches.

Chapter 1

Overview on MHPs

1.1 Crystal structure

A metal halide perovskite is essentially composed of three blocks, a cation which could be inorganic, such as Cs^+, or organic, such as methylammonium $(CH_3NH_3)^+$ [MA] or formamidinium $((NH_2)_2CH)^+$ [FA], a group IV metal (usually Pb or Sn) and an halogen (usually I,Br,Cl).
The spatial arrangement of such molecules is a tilted octahedra with bondings between metal and halogen while the cation sits in the interstices.

1.2 Real world applications

MHPs have been shown to regularly reach efficiencies beyond 20% (23% in the best cases) [5] which is comparable to traditional silicon based cells and just slightly worse than new cutting edge GaAs cells (28%). Where MHPs truly shine though is scalability, regarding both material harvesting and production methods.
As stated by Henry J Snaith in his 2018 paper "Present status and future prospects of perovskite photovoltaics" [2]

> "In order to reach 12.5 TW photovoltaic electricity generation, GaAs PV cells would require 500 years of today's gallium production capacity, thin film CdTe photovoltaic modules would require 1,000 years of today's tellurium, and thin film copper indium gallium selenide (CIGS) would require 400 years of today's indium. The only commercial photovoltaic technology that can scale to this level is silicon which [...] would require around three years of our current silicon production. In stark contrast, perovskites would only take a few days of current lead production capacity to scale to the TW level."

Production of MHPs is also remarkably easy given that the most popular method of producing MHP films is through low temperature depositing of solutions containing the basic elements of such crystals.

1.3 Limiting factors

The two biggest challenges MHPs have to face before hitting the mainstream market are long term stability and environmental sustainability.

Addressing stability first, modern MHPs based cells can retain a satisfying efficiency for more than 1000 hours of operations (or 42 days approximately) and are close to meeting the requirements of the US Electrotechnical Commission (IEC) 61215 industrial standard. It is however believed that, in order to pose as a suitable candidate for real world energy production and thus, attracting large scale funding efficiency up to 25 years is expected.
One of the reasons for such a low stability is, unfortunately, believed to be the very property that

makes MHPs so easy to produce, that being ionic bonding. The fact that PVs are easily synthesized at low temperatures means that at the same temperatures bonding degradation and recombination are certainly factors in play. [2]

Regarding eco sustainability the results are still unclear, the biggest concern being the use of lead in most stable compounds. While it is believed that global lead production would not need to increase significantly in order to meet photovoltaic production requirements more thorough life cycle assessments (LCA) are being conducted in order to fully understand the impact of lead based MHPs on the environment. At the same time research and production of lead free perovskites is already underway with the only suitable candidate efficiency wise being tin (Sn) but stability is even more of an issue with acceptable compromises coming from hybrid lead - tin MHPs.

Chapter 2

General analysis

As stated in the previous section, the three components of a MHP are the cation, the metal and the halide. Using data from previous publications [8] [4] [1] [3] the objective is to study the behaviour of Energy Gap and effective masses varying the three factors. While several simulation methods have been conducted (namely, SR-DFT, SR-GW, SOC-DFT, SOC-GW) the latter is the only one that has regularly shown a satisfying agreement with experimental data and thus, will be the one presented.

2.1 Changing the metal

In this section the behaviour of pure and hybrid Pb/Sn based MHPs is evaluated. [8] [4]
The general formula for such compound is

$$MASn_xPb_{1-x}I_3$$

With $0 \leq x \leq 1$ representing the fraction of tin in the compound.
Starting from the pure Pb based perovskite (x=0) the Sn content is progressively increased until a pure Sn perovskite is achieved (x=1).
For x=0,25 (3 Pb, 1 Sn) and x=0,75 (1 Pb, 3 Sn) four redundant structures are formed while for x=0,50 there are three non equivalent structures based on the positions of the atoms, however further research has shown negligible difference between the structures so the potential problem does not arise.

x	0.00	0.25	0.50	0.75	1.00
$Vol[Å^3]$	976	972	966	959	943
$\Delta Vol(\%)$	0.0	-0.4	-1.0	-1.8	-3.3
$E_g[eV]SOC-GW$	1.64	1.37	1.41	1.20	1.31
$E_g[eV]EXP-A$	1.51	1.31	1.28	1.23	1.10
$E_g[eV]EXP-B$	1.55	1.24	1.17	1.17	1.30

Going towards the lighter metal a slight decrease in cell volume is noticed together with a decrease in the band gap. For reference effective hole and electron masses are shown for the two pure PVs

SOC-GW	$MAPbI_3$	$MASnI_3$
$m_h[a.u.]$	0.25	0.13
$m_e[a.u.]$	0.15	0.17

2.2 Changing the halide

In this section the behaviour of the $MAPbX_3$ perovskite is evaluated, with X=Cl^{17},Br^{35},I^{53}. [3]

x	I	Br	Cl
$Vol(A^3)$	976	816	719
$\Delta Vol(\%)$	0.0	-16	-26
$E_g[eV]SOC-GW$	1.67	2.56	3.46
$E_g[eV]EXP$	1.60	2.35	3.46
$m_h[a.u.]$	0.33	0.31	0.32
$m_e[a.u.]$	0.23	0.27	0.32

In this case the decrease in cell volume is very significant together with an increase in band gap and electron effective mass which heavily favours the heavier halides regarding electron mobility (see next chapter).
Hole mass is not significantly affected.

2.3 Changing the cation

In this section the behaviour of $APbI_3$ perovskites are evaluated, with A being MA $(CH_3NH_3)^+$ or FA $((NH_2)_2CH)^+$. [1]

SOC-GW	$MAPbI_3$	$FAPbI_3$
$Vol(A^3)$	977	981
$\Delta Vol(\%)$	0.0	0.4
$E_g[eV]SOC-GW$	1.67	1.48
$E_g[eV]EXP$	1.66	1.58
$m_h[a.u.]$	0.25	0.23
$m_e[a.u.]$	0.19	0.18

Along with a negligible increase in cell volume a significant decrease in band gap and effective masses is noticed, which in turn favours mobility. It has indeed been show how larger cations are to be preferred even though the improvements are non linear with size and bigger cations lead to progressively diminished returns. [1]

Chapter 3

Simulations on $MAPbI_3$

To better understand the factors in play regarding MHP mobilities a sequence of simulations are performed using the tethragonal $MAPbI_3$ as a benchmark. Through the software "Easy Mobility" [6] made for calculating electron and hole mobility one parameter at the time is changed and compared to the reference values for the particular crystal.
The parameters which are going to be modified are
- hole effective mass
- electron effective mass
- high frequency dielectric constant
- energy gap
- temperature

Based on previous analysis of the compound [8], reference values for such parameters are set to

$m_{h(SOC-GW)}$ [a.u.]	0.25
$m_{e(SOC-GW)}$ [a.u.]	0.19
$E_{g(SOC-GW)}$ [eV]	1.67
ϵ_∞	7.08
T[K]	300

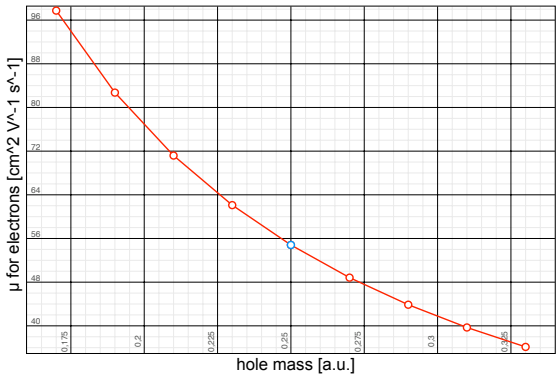
Running the program with those parameters gives the results

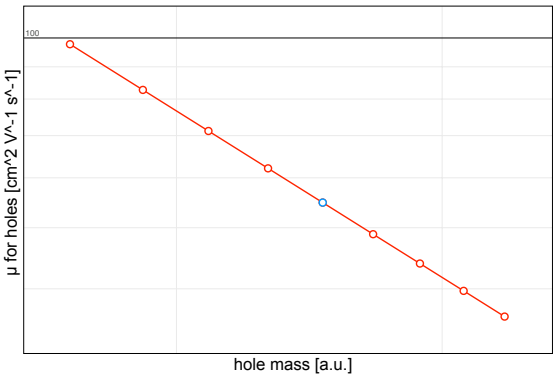
$\mu_h[cm^2V^-1s^-1]$	54.81
$\mu_e[cm^2V^-1s^-1]$	82.74

Which are going to be compared with the new values that will be extracted in the following simulations.
NOTE: in all the following plots the odd coloured dot is the reference value.

3.1 Hole effective mass

$m_h[a.u.]$	$\mu_h[cm^2V^-1s^-1]$
0.19	97.74
0.19	82.72
0.21	71.19
0.23	62.11
0.25	54.81
0.27	48.83
0.29	43.87
0.31	39.70
0.33	36.14





In the second graph with a log-log scale a clear dependence arises in the form of

$$\log(\mu_h) = A\log(m_h) + B \quad \rightarrow \quad \mu_h = e^B m_h^A = C m_h^A$$

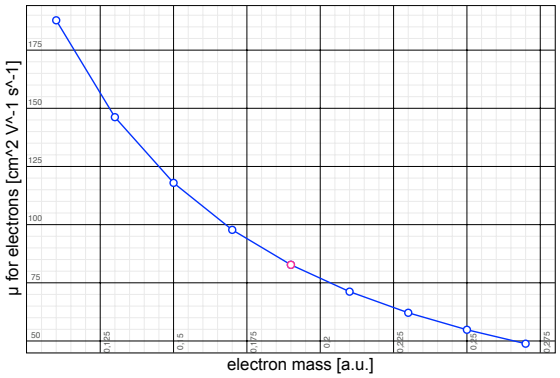
Such parameters are extracted through an interpolation

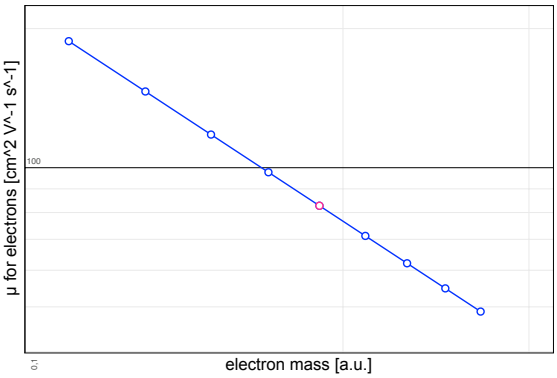
	Holes
A	-1.50 ± 0.03
C	6.9 ± 0.3

Electron mobility is not affected by hole mass

3.2 Electron effective mass

$m_h[a.u.]$	$\mu_h[cm^2V^-1s^-1]$
0.11	187,82
0.13	146.19
0.15	117.95
0.17	97.76
0.19	82.74
0.21	71.20
0.23	62.12
0.25	54.82
0.27	48.84





Following the same reasoning as in the previous section:

$$\log(\mu_e) = A\log(m_e) + B \rightarrow \mu_e = e^B m_e^A = C m_e^A$$

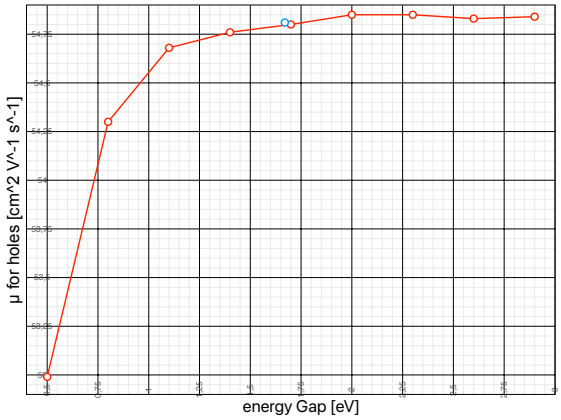
	Electrons
A	-1.50 ± 0.02
C	6.9 ± 0.2

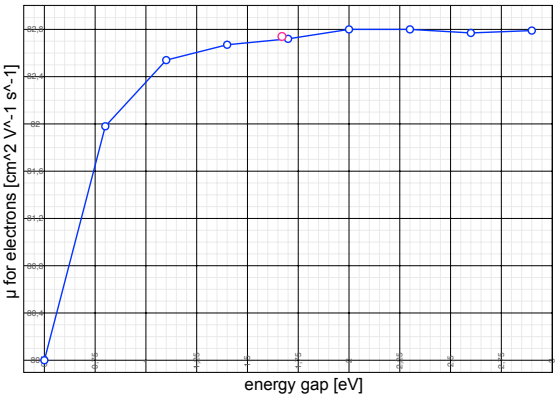
Hole mobility is not affected by electron mass.

It's interesting to note how the behaviour of holes and electrons is perfectly symmetrical when changing their respective effective mass values.

3.3 Energy gap

$E_g[eV]$	$\mu_h[cm^2V^-1s^-1]$	$\mu_e[cm^2V^-1s^-1]$
0.5	52.99	80.00
0.8	54.30	81.98
1.1	54.68	82.54
1.4	54.76	82.67
1.7	54.80	82.72
2.0	54.85	82.80
2.3	54.85	82.80
2.6	54.83	82.77
2.9	54.84	82.79



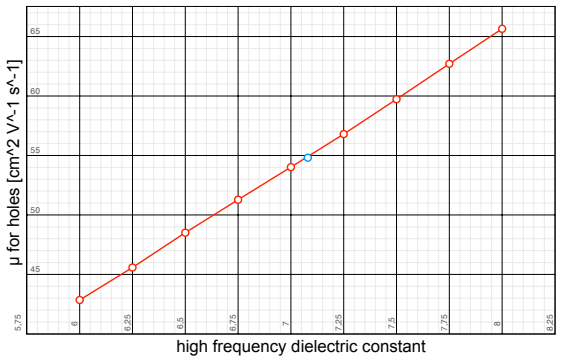


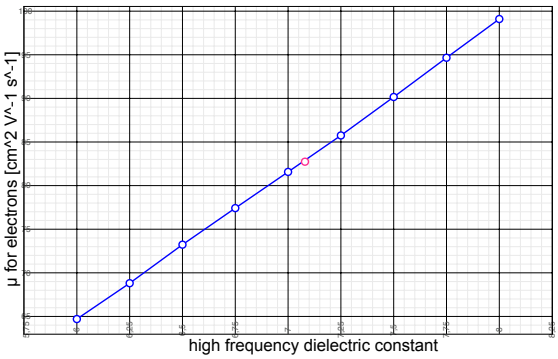
In this instance a clear analytic relation is harder to find but, interestingly, for $E_g \geq 1.5eV$ the energy gap becomes almost uninfluential to the result of the mobility with a saturation value of

$$\mu_h^\infty \approx 54.85 \text{ and } \mu_e^\infty \approx 82.78$$

3.4 Dielectric constant

ϵ_∞	$\mu_h[cm^2V^-1s^-1]$	$\mu_e[cm^2V^-1s^-1]$
6.00	42.85	64.70
6.25	45.58	68.81
6.50	48.51	73.23
6.75	51.28	77.42
7.00	54.02	81.57
7.75	56.80	85.75
7.50	59.73	90.16
7.75	62.71	94.66
8.00	65.65	99.10





The behaviour here is clearly linear so

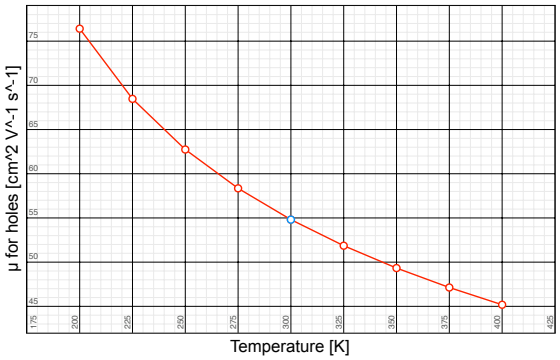
$$\mu_{e/h} = A_{e/h}\epsilon_\infty + B_{e/h}$$

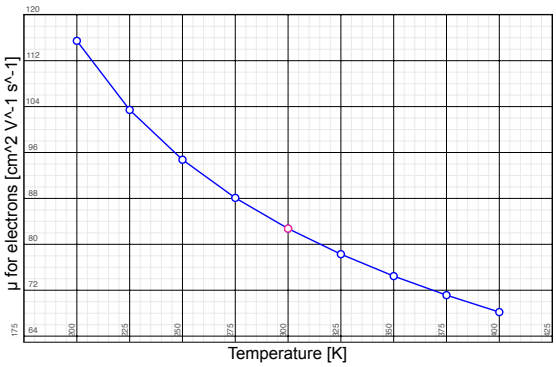
By performing linear fits the results are

	Holes	Electrons
A	11.37 ± 0.05	17.16 ± 0.09
B	-25.5 ± 0.4	-38.4 ± 0.6

3.5 Temperature

$T[K]$	$\mu_h[cm^2V^-1s^-1]$	$\mu_e[cm^2V^-1s^-1]$
200	76.41	115.46
225	68.47	103.42
250	62.74	94.74
275	58.35	88.09
300	54.81	82.74
325	51.86	78.28
350	49.33	74.46
375	47.13	71.74
400	45.18	68.19





The decrease in mobility while increasing temperature is to be expected since less thermal agitation leads to less scattering and, thus, increased mobility.

Chapter 4

Vibrational analysis

4.1 Method overview

The last section of the thesis is an analysis on the vibrational contribution of different atoms to infrared phonon scattering and, thus, MHP mobility. Through a custom written C++ software (which can be found in the appendix) it's possible to eliminate vibrational modes for certain atoms in the crystal. The main focus of the software is to compute the expression [7]

$$F_n^2 = \sum_{i=1}^{3}(\sum_{I=1}^{N_{at}}\sum_{j=1}^{3}\frac{4\pi}{3V}\frac{Z_{iIj}u_{ij}^n}{\sqrt{M_I}})^2$$

Where N_{at} is the number of atoms of the studied cell (composed by four copies of the base cell), n runs over the vibrational modes, in this study set to be $n = 1, 2, ..., 3N_{at}$, i runs over the polarization directions and j over the Cartesian directions.
Z_{iIj} is the born effective charge tensor and it is given as an input.
Before being using Z_{iIj} in the computation a correction is needed, called "acoustic sum rule". This stems from the fact that, being the crystal electrically neutral, the sum of all forces must equal 0, this doesn't happen in simulations due to various numerical error thus the correction is needed.
u^n is the normalised phonon eigenmode of frequency ω_n (O.N.). In input we however have slightly different v^n vectors which are still normalised but need to be corrected by a factor

$$u_I^n = v_I^n * \sqrt{M_I} \quad \text{with} \quad I = 1, .., N_{at}$$

And then re normalised in order to give a proper orthonormal set.
The software will now compute F_n^2.
At this point the software will give the option to delete the contribution of certain atoms (chosen by name or by their index in the crystal) by acting on the vectors u^n. To be more accurate, for any selected set of indexes $\tilde{I} \subset I = 1, ..., N_{at}$

$$u_i^n = 0 \qquad \forall i \in \tilde{I}$$

The next step is the computation of $\eta^n = \frac{<\tilde{u_I^n}|u_I^n>}{<u_I^n|u_I^n>}$ with $\tilde{u_I^n}$ the edited vibration vectors and correct the oscillation strengths $\tilde{F}_n^2 = \eta^n F_n^2$
Corrected $\tilde{F}_n^2$ can now be given as an input for the software "Easy Mobility" [6].

4.2 Simulations

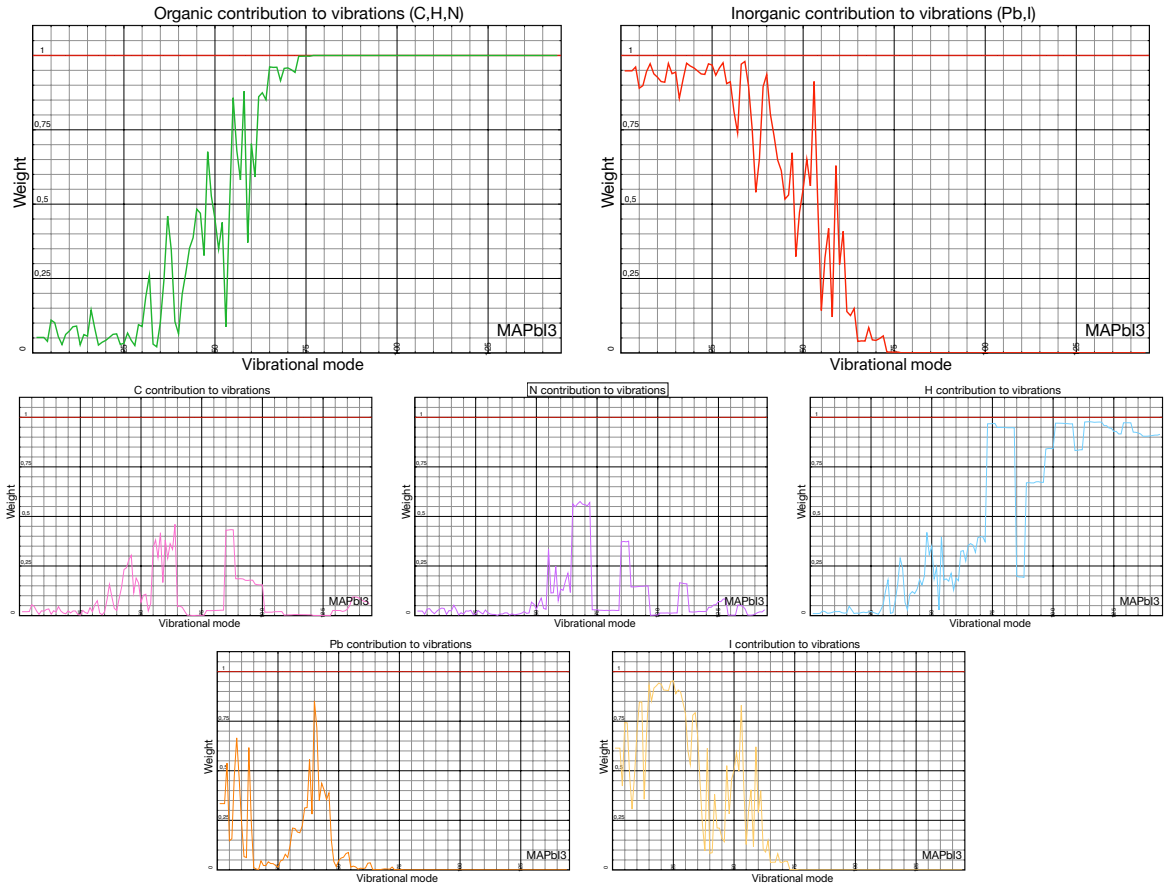
Three types of compound have been analysed. The aforementioned $MAPbI_3$, $FAPbI_3$ and the hybrid $FAMAPbI_3$.
For all compounds the same case scenarios have been evaluated:
- Deactivated halide
- Deactivated metal
- Deactivated metal and halide
- Deactivated organic cation

Along with mobility calculations the contribution of each atom (or group of) to any vibrational mode has been computed and plotted, the weight being non other than η^n for the particular configuration.

4.2.1 $MAPbI_3$

$MAPbI_3$	$\mu_h[cm^2V^-1s^-1]$	$\mu_e[cm^2V^-1s^-1]$
Original	49.45	74.80
No Halide	59.10	89.20
No Metal	51.64	77.96
No Metal and Halide	92.18	150.88
No Organic	50.82	76.71

The mobility increase in every configuration compared to the original is to be expected since less vibrations lead to less IR scattering. The main part of the scattering seems to come from the halide in conjunction with the metal since neither metal or organic deactivation alone lead to significant increases in mobility.



Regarding atom contributions it's interesting to note how low frequency modes ($n \leq 30$) are almost entirely driven by the inorganic part of the crystal, there is then a transient phase for $30 \leq n \leq 75$ in which both section of the atom contribute followed by high frequency modes in which only the organic part is significantly present.
Looking at singular atomic species the trend is confirmed, with Pb^{82}, I^{53} contributing for low frequency

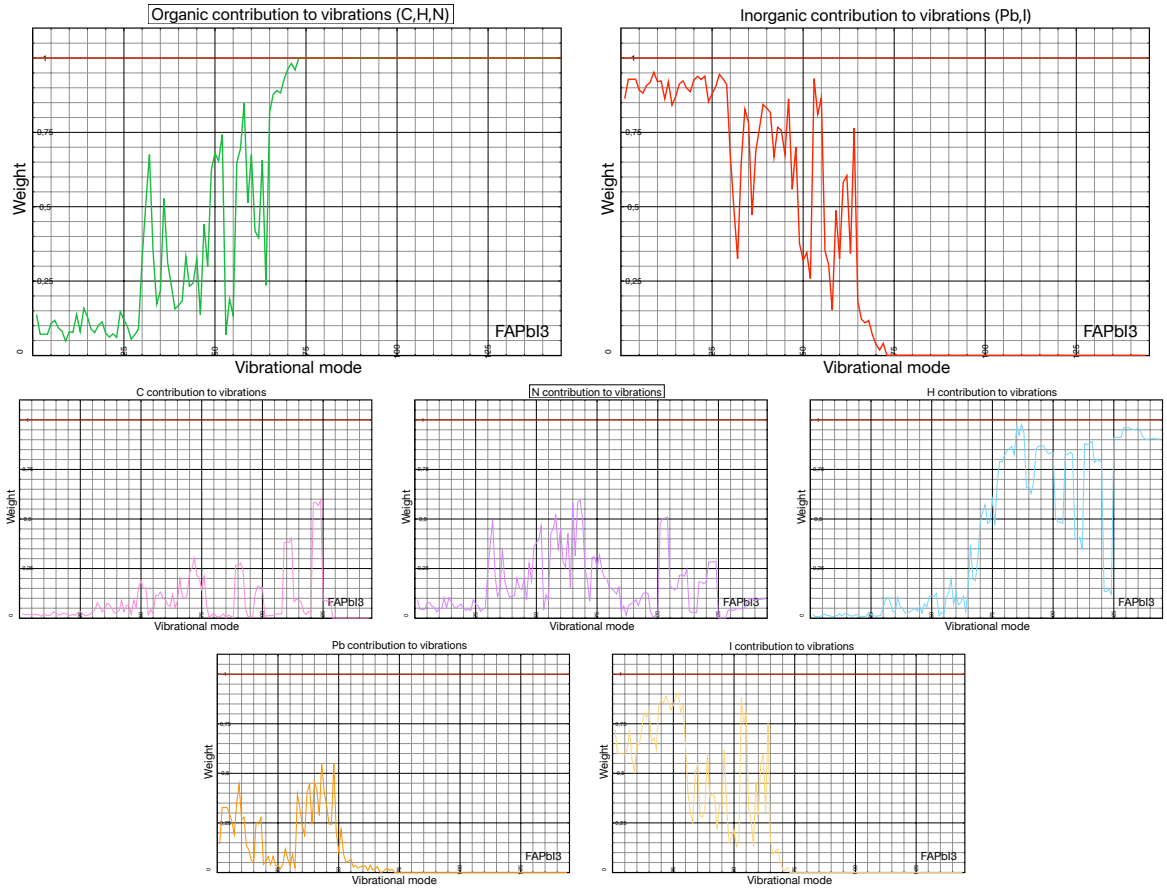
modes, N^7, C^6 for mid frequencies and H^1 for high frequencies, which could suggest a connection between atomic weight and vibration frequency (heavier atoms contribute more to slow oscillations). In conclusion, the simulations show that the majority of the IR scattering comes from the inorganic part of the crystal which, in turn, is responsible for low frequency modes in IR phonon scattering.

NOTE: In regards to chapter 3 the different mobility value stems from the usage of an improved data set not available at the time of the first simulations.

4.2.2 $FAPbI_3$

The trend established for the previous compound is confirmed for both mobilities and atom contributions.

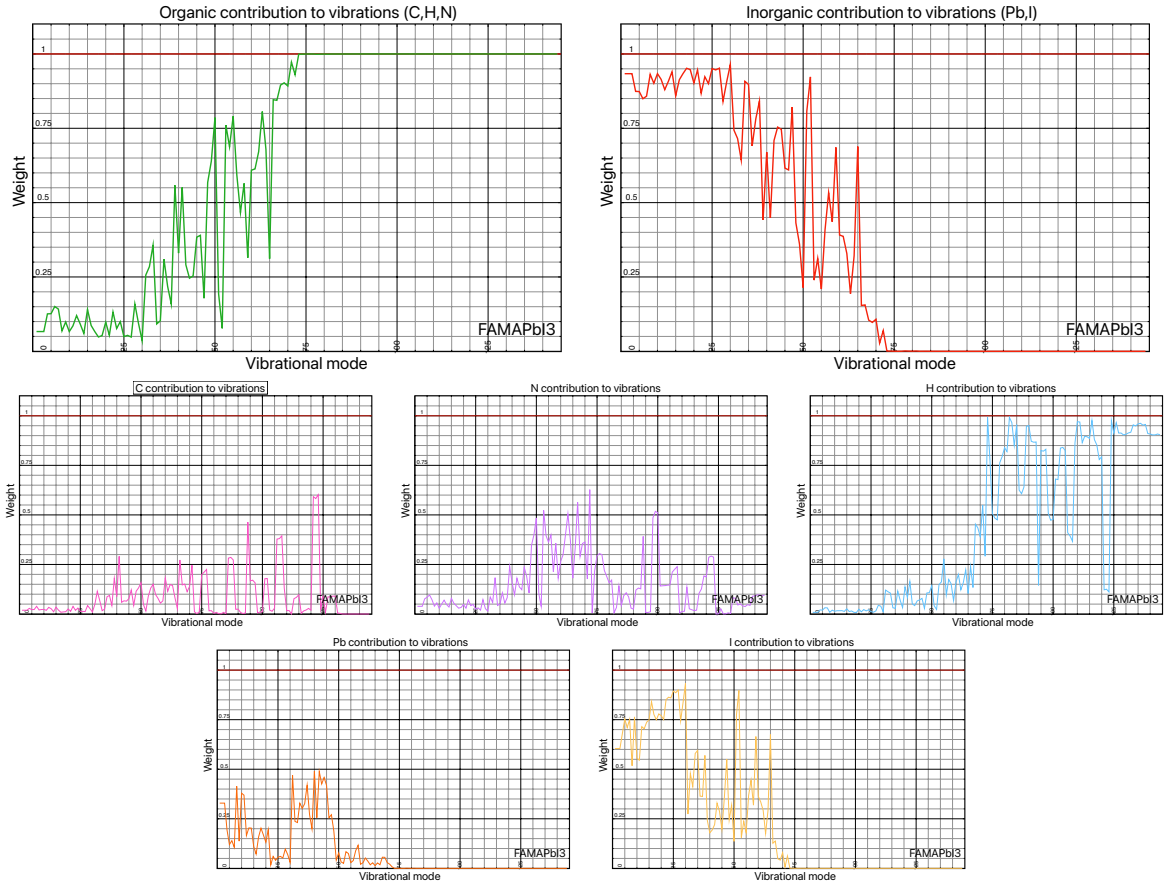
$FAPbI_3$	$\mu_h[cm^2V^-1s^-1]$	$\mu_e[cm^2V^-1s^-1]$
Original	63.67	91.98
No Halide	78.35	113.18
No Metal	70.07	101.23
No Metal and Halide	111.95	161.97
No Organic	66.89	96.62



4.2.3 $FAMAPbI_3$

The trend established for the previous compounds is confirmed for both mobilities and atom contributions.

$FAMAPbI_3$	$\mu_h[cm^2V^-1s^-1]$	$\mu_e[cm^2V^-1s^-1]$
Original	57.17	84.50
No Halide	71.11	105.08
No Metal	62.66	92.60
No Metal and Halide	98.13	145.00
No Organic	60.80	89.85



Chapter 5

Conclusions

In this thesis work a thorough evaluation of the main variables regarding MHPs mobility has been performed.
Starting from intrinsic parameters such as effective masses, energy gap, dielectric constant and external conditions like the temperature of operation the focus has been shifted towards infrared phonon scattering.
In the last round of simulation an exceptional contribution of low frequency vibration to the scattering has been highlighted which, in turn, is mainly tied to the inorganic part of the crystal while the organic part plays a very minor role.
The analysis on the factors contributing to the mobility of MHPs sets a foundation for future studies, especially regarding low frequency contributions to IR scattering which could lead to the development of new and improved materials should valuable alternatives to Pb and I arise.

Appendix A

Code

```
#include <iostream>
#include <vector>
#include <cmath>
#include <string>
#include <fstream>
#include <sstream>
using namespace std;

int main()
{
  bool x;
  string vib, line, bec, mass, check;
  int cellsize, n = 0, I = 0, j = 0;
  cout << "Input cell size \n";
  cin >> cellsize;
  cout << "Input vibrations file (.txt) \n";
  cin >> vib;
  cout << "Input masses file (.txt) \n";
  cin >> mass;
  cout << "Input Born Effective Charge file (.txt) \n";
  cin >> bec;
  cout << "Input reference file for Harmonic Oscillator Strenght (.txt) \n";
  cin >> check;

  double vibrations[3 * cellsize][3 * cellsize];
  double masses[cellsize];
  string labels[cellsize];
  double borneff[cellsize][3][3];
  double oscstr[3][3 * cellsize];
  double ref[3 * cellsize];
  double delta[3 * cellsize];
  double HOS [3 * cellsize];

  ifstream myfile(vib);
  if (myfile.is_open())
  {
    while (getline(myfile, line))
    {
      stringstream iss(line);
```

```
    double a, b, c, d, e, f;
    if (!(iss >> a >> b >> c >> d >> e >> f))
    {
      break;
    }
    vibrations[I][j] = a;
    vibrations[I + 1][j] = c;
    vibrations[I + 2][j] = e;
    I = I + 3;

    if (I == 3 * cellsize)
    {
      I = 0;
      j++;
    }
  }
  myfile.close();
}
else
{
  cout << "Unable to open file " << vib;
  return 1;
}

cout << "Do you want to check the file " << vib << " ? [Y=1;N=0] \n";
cin >> x;
if (x == 1)
{
  cout << "CHECK INPUT FOR FILE " << vib << '\n';
  for (int i = 0; i < 3 * cellsize; i++)
  {
    cout << "vibrational mode " << i + 1 << '\n';
    for (int n = 0; n < cellsize; n++)
    {
      cout << vibrations[3 * n][i] << '\t' << vibrations[3 * n + 1][i] <<
      '\t' << vibrations[3 * n + 2][i] << '\n';
    }
  }
}

j = 0;

ifstream ourfile(mass);
if (ourfile.is_open())
{
  while (getline(ourfile, line))
  {
    stringstream iss(line);
    double a, c;
    string b;
    if (!(iss >> a >> b >> c))
    {
      break;
```

```
    }
    labels[j] = b;
    masses[j] = c;
    j++;
  }
}
else
{
  cout << "Unable to open file " << mass;
  return 1;
}

cout << "Do you want to check the file " << mass << " ? [Y=1;N=0] \n";
cin >> x;
if (x == 1)
{
  cout << "CHECK INPUT FOR FILE " << mass << '\n';
  for (int n = 0; n < cellsize; n++)
  {
    cout << labels[n] << '\t' << masses[n] << '\t' << sqrt(masses[n]) << '\n';
  }
  cout << '\n';
}

j = 0;
I = 0;

ifstream themfile(bec);
if (themfile.is_open())
{

  while (getline(themfile, line))
  {
    stringstream iss(line);
    double a, b, c;
    if (!(iss >> a >> b >> c))
    {
      break;
    }
    borneff[I][j][0] = a;
    borneff[I][j][1] = b;
    borneff[I][j][2] = c;
    j++;
    if (j == 3)
    {
      j = 0;
      I++;
    }
  }
}
else
{
  cout << "Unable to open file " << bec;
```

```
  return 1;
}

for (int i = 0; i < 3; i++)
{
  for (int j = 0; j < 3; j++)
  {
    double sum = 0;
    for (int n = 0; n < cellsize; n++)
    {
      sum += borneff[n][i][j];
    }
    for (int n = 0; n < cellsize; n++)
    {
      borneff[n][i][j] -= (sum / cellsize);
    }
  }
}

cout << "BEC tensors corrected for acoustic sum rule \n";

cout << "Do you want to check the file " << bec << " ? [Y=1;N=0] \n";
cin >> x;
if (x == 1)
{
  cout << "CHECK INPUT FOR FILE " << vib << '\n';
  for (int i = 0; i < cellsize; i++)
  {
    cout << "BEC tensor for atom " << i + 1 << '\n';
    for (int n = 0; n < 3; n++)
    {
      cout << borneff[i][n][0] << '\t' << borneff[i][n][1] <<
      '\t' << borneff[i][n][2] << '\n';
    }
  }
}

ifstream hefile(check);
if (hefile.is_open())
{
  while (getline(hefile, line))
  {
    stringstream iss(line);
    double a, b, c, d;
    if (!(iss >> a >> b >> c >> d))
    {
      break;
    }
    ref[j] = d;
    j++;
  }
}
else
```

```
{
  cout << "Unable to open file " << check;
  return 1;
}

cout << "Correcting and renormalising vibration vectors \n";
cout << "Do you want to check the renormalisation? [Y=1;N=0] \n";
cin >> x;

for (int i = 0; i < 3 * cellsize; i++)
{

  double lenght = 0;
  for (int n = 0; n < cellsize; n++)
  {

    double k = vibrations[3 * n][i] * vibrations[3 * n][i] +
               vibrations[3 * n + 1][i] * vibrations[3 * n + 1][i] +
               vibrations[3 * n + 2][i] * vibrations[3 * n + 2][i];

    lenght += k;
  }
  double norm = sqrt(lenght);

  if (x == 1)
  {
    cout << "The norm of v_In as input for the mode " << i + 1 << " is " << norm << '\n';
  }

  for (int n = 0; n < cellsize; n++)
  {
    // cout << sqrt(masses[n]) << '\n';
    vibrations[3 * n][i] = vibrations[3 * n][i] * sqrt(masses[n]);
    vibrations[3 * n + 1][i] = vibrations[3 * n + 1][i] * sqrt(masses[n]);
    vibrations[3 * n + 2][i] = vibrations[3 * n + 2][i] * sqrt(masses[n]);
  }

  double lenght2 = 0;
  for (int n = 0; n < cellsize; n++)
  { // indice n scorre gli atomi

    double k = vibrations[3 * n][i] * vibrations[3 * n][i] +
               vibrations[3 * n + 1][i] * vibrations[3 * n + 1][i] +
               vibrations[3 * n + 2][i] * vibrations[3 * n + 2][i];

    lenght2 += k;
  }
  double norm2 = sqrt(lenght2);

  if (x == 1)
  {
    cout << "The norm of the vector u_In= v_In * sqrt(M_I) for the mode" <<
    i + 1 << " is " << norm2 << '\n';
```

```
  }

  for (int n = 0; n < cellsize; n++)
  {
    vibrations[3 * n][i] /= norm2;
    vibrations[3 * n + 1][i] /= norm2;
    vibrations[3 * n + 2][i] /= norm2;
  }

  double lenght3 = 0;
  for (int n = 0; n < cellsize; n++)
  { // indice n scorre gli atomi

    double k = vibrations[3 * n][i] * vibrations[3 * n][i] +
               vibrations[3 * n + 1][i] * vibrations[3 * n + 1][i] +
               vibrations[3 * n + 2][i] * vibrations[3 * n + 2][i];

    lenght3 += k;
  }

  double norm3 = sqrt(lenght3);

  if (x == 1)
  {
    cout << "The renormalised u_In norm for the mode" <<
    i + 1 << " is " << norm3 << '\n';
  }
}

cout << "Do you want to ckeck orthogonality for u_n? [Y=1;N=0] \n";
cin >> x;

for (int i = 0; i < 3 * cellsize; i++)
{

  int j = 3 * cellsize - 1;
  while (j > i)
  {
    double ort = 0;
    for (int n = 0; n < cellsize; n++)
    {
      double diff = vibrations[3 * n][i] * vibrations[3 * n][j] +
                    vibrations[3 * n + 1][i] * vibrations[3 * n + 1][j] +
                    vibrations[3 * n + 2][i] * vibrations[3 * n + 2][j];
      ort += diff;
    }

    if (x == 1)
    {
      cout << "Mode " << i + 1 << " scalar " << j + 1 << " = " << ort << '\n';
    }

    j = j - 1;
```

```
  }
}

cout << "Computing Harmonic Oscillator Strenghts \n";

for (int i = 0; i < 3 * cellsize; i++)
{
  for (int k = 0; k < 3; k++)
  {
    double s = 0;
    for (int n = 0; n < cellsize; n++)
    {
      for (int j = 0; j < 3; j++)
      {
        s += (vibrations[3 * n + j][i] * borneff[n][k][j]) / sqrt(masses[n]);
      }
    }
    oscstr[k][i] = s;
  }
  HOS[i] = pow(oscstr[0][i], 2) + pow(oscstr[1][i], 2) + pow(oscstr[2][i], 2);
  cout << "HOS for mode " << i + 1 << " is " << HOS[i] << '\n';
}

cout << "Do you want to calculate the corrective factor or do you want to
manually input it? [1/0] \n";
cin >> x;
double mean = 0;
if (x == 1)
{
  double s = 0;
  for (int i = 0; i < 3 * cellsize; i++)
  {
    double corr = ref[i] / HOS[i];
    s += corr;
    cout << "The corrective factor for the mode " << i + 1 << " is " << corr << '\n';
  }
  mean = s / (3 * cellsize - 3);
  cout << "The average corrective factor is " << mean << '\n';
}

if (x == 0)
{
  cout << "Input corrective factor \n";
  cin >> mean;
}

cout << "Do you want to disable vibrations for certain atoms? [Y=1;N=0] \n";
bool y;
cin >> x;
if (x == 1)
{

  cout << "Do you want to chose by atom name or by atom index inside the crystal?
```

```
[1=name;0=index] \n";
cin >> y;
if (y == 1)
{
  string name;
  int I = 0;
  while (I < 1)
  {
    cout << "Input atom name, type 'out' to exit the selection \n";
    cin >> name;
    if (name == "out")
    {
      I++;
    }
    for (int n = 0; n < cellsize; n++)
    {
      if (labels[n] == name)
      {
        for (int i = 0; i < 3 * cellsize; i++)
        {
          vibrations[3 * n][i] = 0.;
          vibrations[3 * n + 1][i] = 0.;
          vibrations[3 * n + 2][i] = 0.;
        }
        cout << "Deleted vibrations for the atom " << labels[n] <<
        " with index " << n + 1 << '\n';
      }
    }
  }
  cout << "Do you want to check the edited vibrations file? [Y=1;N=0] \n";
  cin >> x;
  if (x == 1)
  {
    cout << "CHECK INPUT FOR FILE " << vib << '\n';
    for (int i = 0; i < 3 * cellsize; i++)
    {
      cout << "vibrational mode " << i + 1 << '\n';
      for (int n = 0; n < cellsize; n++)
      {
        cout << vibrations[3 * n][i] << '\t' << vibrations[3 * n + 1][i] << '\t' <<
        vibrations[3 * n + 2][i] << '\n';
      }
    }
  }
}

if (y == 0)
{
  int n;
  int I = 0;
  while (I < 1)
  {
    cout << "Input atom index [1;" << cellsize << "], type '0' to exit the selection \n";
```

```
      cin >> n;
      bool y = 1;
      if (n == 0)
      {
        y = 0;
        I++;
      }
      if (y == 1)
      {
        for (int i = 0; i < 3 * cellsize; i++)
        {
          vibrations[3 * (n - 1)][i] = 0.;
          vibrations[3 * (n - 1) + 1][i] = 0.;
          vibrations[3 * (n - 1) + 2][i] = 0.;
        }
        cout << "Deleted vibrations for the atom " << labels[n - 1] <<
        " with index " << n << '\n';
      }
    }
    cout << "Do you want to check the edited vibrations file? [Y=1;N=0] \n";
    cin >> x;
    if (x == 1)
    {
      cout << "CHECK INPUT FOR FILE " << vib << '\n';
      for (int i = 0; i < 3 * cellsize; i++)
      {
        cout << "vibrational mode " << i + 1 << '\n';
        for (int n = 0; n < cellsize; n++)
        {
          cout << vibrations[3 * n][i] << '\t' << vibrations[3 * n + 1][i] <<
          '\t' << vibrations[3 * n + 2][i] << '\n';
        }
      }
    }
  }
}

cout << "Computing delta correction \n";
for (int i = 0; i < 3 * cellsize; i++)
{
  double norm = 0;
  for (int n = 0; n < cellsize; n++)
  {
    norm += vibrations[3 * n][i] * vibrations[3 * n][i] +
            vibrations[3 * n + 1][i] * vibrations[3 * n + 1][i] +
            vibrations[3 * n + 2][i] * vibrations[3 * n + 2][i];
  }
  delta[i] = norm;
}
cout << "Do you want to check the delta corrections? [Y=1;N=0] \n";
cin >> x;
if (x == 1)
{
```

```
    for (int i = 0; i < 3 * cellsize; i++)
    {
      cout << "vibrational mode " << i + 1 << '\t' << delta[i] << '\n';
    }
  }

  cout << "Corrected HOS (with delta and corrective factor) are \n";

  for (int i = 0; i < 3 * cellsize; i++)
  {
    HOS[i] = HOS[i] * delta[i] * mean;
    cout << "Mode " << i + 1 << '\t'<< HOS[i] << '\n';
  }

  return 0;
}
```

Bibliography

[1] Anna Amat et al. "Cation-Induced Band-Gap Tuning in Organohalide Perovskites: Interplay of Spin–Orbit Coupling and Octahedra Tilting". In: *Nano Letters* 14.6 (2014). PMID: 24797342, pp. 3608–3616. DOI: 10.1021/nl5012992. eprint: https://doi.org/10.1021/nl5012992. URL: https://doi.org/10.1021/nl5012992.

[2] Snaith H.J. "Present status and future prospects of perovskite photovoltaics". In: *Nature Mater* 17 (2018), pp. 372–376. DOI: https://doi.org/10.1038/s41563-018-0071-z.

[3] Edoardo Mosconi, Paolo Umari, and Filippo De Angelis. "Electronic and optical properties of MAPbX3 perovskites (X = I, Br, Cl): a unified DFT and GW theoretical analysis". In: *Phys. Chem. Chem. Phys.* 18 (39 2016), pp. 27158–27164. DOI: 10.1039/C6CP03969C. URL: http://dx.doi.org/10.1039/C6CP03969C.

[4] Edoardo Mosconi, Paolo Umari, and Filippo De Angelis. "Electronic and optical properties of mixed Sn–Pb organohalide perovskites: a first principles investigation". In: *J. Mater. Chem. A* 3 (17 2015), pp. 9208–9215. DOI: 10.1039/C4TA06230B. URL: http://dx.doi.org/10.1039/C4TA06230B.

[5] "Perovskites keep on giving". In: *Nature Mater* 17 (2018), p. 371. DOI: https://doi.org/10.1038/s41563-018-0076-7.

[6] Paolo Umari. *Easy Mobility*. URL: https://gitlab.com/paoloumari/easy-mobility.

[7] Paolo Umari. "Inclusion of infrared dielectric screening in the GW method from polaron energies to charge mobilities". In: *NPJ Comput. Materials* (2022). DOI: 10.1038/s41524-022-00812-9. URL: https://doi.org/10.1038/s41524-022-00812-9.

[8] Paolo Umari, Edoardo Mosconi, and Filippo De Angelis. "Relativistic GW calculations on CH3NH3PbI3 and CH3NH3SnI3 Perovskites for Solar Cell Applications". In: *Sci Rep* 4 (2014), p. 4467. DOI: https://doi.org/10.1038/srep04467.