

pymrio-tutorial-for-wiod

January 18, 2018

1 Quick Start Pymrio Tutorial using WIOD

This notebook contains the interactive version of the quick start given in the Pymrio article (Stadler et al 2018 sub).

Pymrio requires a Python version ≥ 3.5 . If you don't have Python installed, I recommend to use the [Anaconda Scientific Python package](#).

Pymrio is available on [the Python Package Index PyPI](#) and on the [Anaconda Cloud](#). Thus, two possibilities exist to install Pymrio and all required packages.

For using the version on PyPI use:

```
pip install pymrio --upgrade
```

To install from the Anaconda Cloud do:

```
conda install -c konstantinstadler pymrio
```

You can then import the Pymrio package with

```
In [1]: import pymrio
```

In this example here, we will use the [WIOD MRIO database](#).

First, the Pymrio MRIO download function is used to get the WIOD MRIO database with:

```
In [2]: raw_wiod_path = '/tmp/wiod/raw'
        pymrio.download_wiod2013(storage_folder=raw_wiod_path,
                                years=[2008])
```

```
Out[2]: Description: WIOD metadata file for pymrio
```

```
MRIO Name: WIOD
```

```
System: ixi
```

```
Version: data13
```

```
File: /tmp/wiod/raw/metadata.json
```

```
History:
```

```
20180112 09:23:36 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/water/w
```

```
20180112 09:23:36 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/materia
```

```
20180112 09:23:35 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/land/la
```

```
20180112 09:23:35 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/AIR/AIR
```

```
20180112 09:23:34 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/C02/C02
```

```

20180112 09:23:33 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/EM/EM_ma
20180112 09:23:32 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/EU/EU_ma
20180112 09:23:32 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/SEA/WIOD
20180112 09:23:31 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/update_

```

This downloads the 2008 MRIO table from WIOD. Omitting the year parameter would result getting all years. The function returns a Pymrio meta data object, which gives information about the WIOD version, system (in this case industry by industry) and records about from where the data was received.

To parse the database into a Pymrio object use:

```
In [3]: wiod = pymrio.parse_wiod(raw_wiod_path, year=2008)
```

The available data can be explored by for example:

```
In [4]: wiod.get_sectors()
```

```
Out[4]: Index(['AtB', 'C', '15t16', '17t18', '19', '20', '21t22', '23', '24', '25',
              '26', '27t28', '29', '30t33', '34t35', '36t37', 'E', 'F', '50', '51',
              '52', 'H', '60', '61', '62', '63', '64', 'J', '70', '71t74', 'L', 'M',
              'N', 'O', 'P'],
              dtype='object', name='sector')
```

or

```
In [5]: wiod.get_regions()
```

```
Out[5]: Index(['AUS', 'AUT', 'BEL', 'BGR', 'BRA', 'CAN', 'CHN', 'CYP', 'CZE', 'DEU',
              'DNK', 'ESP', 'EST', 'FIN', 'FRA', 'GBR', 'GRC', 'HUN', 'IDN', 'IND',
              'IRL', 'ITA', 'JPN', 'KOR', 'LTU', 'LUX', 'LVA', 'MEX', 'MLT', 'NLD',
              'POL', 'PRT', 'ROU', 'RUS', 'SVK', 'SVN', 'SWE', 'TUR', 'TWN', 'USA',
              'RoW'],
              dtype='object', name='region')
```

```
In [6]: wiod.Z
```

```
Out[6]: region
sector      AUS
region sector AtB      C      15t16      17t18      19
AUS  AtB      4445.324330  41.919400  15625.681890  536.968630  154.395870
      C        16.277934  3838.070873   189.934275   11.313686    3.253063
      15t16  1049.495726  100.611347   6754.110522   68.387761   19.663697
      17t18   36.908420   43.779214   108.986668  355.675875  102.268333
      19       9.518107   11.289978    28.105965   91.723266   26.373410
      20      36.819811   47.491171    33.027878    7.296923    2.098101
      21t22  115.900527  153.248506  1246.425549   52.200097   15.009220
      23     490.156419  1026.213186   158.889100    9.179523    2.639409
      24    1025.411077   395.338597   254.521411   58.428801   16.800175
      25      79.516332   189.301500   747.545652   34.083913    9.800233
```

	26	38.658080	105.010424	262.224086	5.283083	1.519054
	27t28	212.089806	1310.988269	836.157336	42.125453	12.112433
	29	168.914698	614.130627	121.573026	9.601957	2.760879
	30t33	100.297729	364.656884	72.187314	5.701425	1.639347
	34t35	94.161797	279.637968	68.165787	8.294429	2.384917
	36t37	52.529450	213.596124	135.683995	26.057759	7.492447
	E	698.423156	1719.046355	885.618629	86.361355	24.831684
	F	633.714316	4242.094785	376.754403	28.695921	8.251007
	50	1132.348543	2138.419658	1023.262884	159.073383	45.738761
	51	1394.463193	1572.454263	4088.518869	419.871502	120.726663
	52	1445.528277	1659.816747	3579.884070	371.567853	106.837791
	H	140.934734	216.771700	561.956674	27.457908	7.895041
	60	476.976916	3184.818553	1993.405269	126.355111	36.331187
	61	9.552743	170.427724	31.177072	4.611883	1.326064
	62	6.869618	20.121560	11.067713	1.786162	0.513581
	63	297.699414	738.735542	1195.706686	45.976713	13.219798
	64	223.155110	423.626089	355.814018	35.236625	10.131668
	J	1299.842059	4293.057285	967.646850	59.138767	17.004318
	70	551.797175	1786.808419	797.474903	32.553813	9.360274
	71t74	1180.063104	5310.446524	3035.907428	368.573815	105.976926
...	
Row	20	1.635056	2.013566	0.820348	0.216980	0.062394
	21t22	7.724918	13.129569	217.877677	2.003884	0.576182
	23	388.152158	836.374409	81.538931	2.390291	0.687287
	24	365.270981	132.696731	32.705396	14.943534	4.296752
	25	8.286214	22.575945	90.951788	2.964467	0.852379
	26	1.956064	6.043732	11.152563	0.099052	0.028483
	27t28	9.581514	191.520139	34.196454	2.872105	0.825822
	29	35.895862	148.406916	4.062407	0.206493	0.059371
	30t33	5.438695	14.663099	4.288701	0.942037	0.270862
	34t35	3.580771	6.292762	1.143145	0.089681	0.025786
	36t37	1.423158	15.919915	2.520675	0.997691	0.286871
	E	0.802647	18.560578	1.105187	0.052474	0.015090
	F	0.690587	4.550382	0.398577	0.018475	0.005314
	50	0.055638	0.158419	0.091477	0.004071	0.001171
	51	2.752223	189.554149	11.612952	0.481597	0.138475
	52	1.031890	1.526614	3.043862	0.176287	0.050688
	H	0.783462	1.108244	3.597303	0.117633	0.033825
	60	0.680310	190.942239	7.225673	0.281982	0.081079
	61	0.085508	0.190833	0.040526	0.003698	0.001063
	62	7.703932	22.053451	12.157162	2.137970	0.614733
	63	0.308886	0.802305	0.698779	0.064745	0.018614
	64	0.166596	0.499486	0.348924	0.032884	0.009460
	J	4.774312	8.175419	8.364439	0.128402	0.036924
	70	7.339967	23.635663	8.385978	0.273455	0.078627
	71t74	3.657849	17.761648	9.827292	1.145630	0.329414
	L	0.547432	0.780406	1.176073	0.200903	0.057766
	M	1.319036	9.927575	11.742183	2.639874	0.759049

N	7.894845	0.291041	11.603507	2.279403	0.655404
O	1.244926	3.686620	4.381357	0.155217	0.044633
P	0.001018	0.000104	0.003666	0.000135	0.000039

region						
sector		20	21t22	23	24	25
region	sector					
AUS	AtB	936.835140	273.018600	0.000000	215.708440	93.909230
	C	14.271582	58.136067	4424.333299	193.328895	20.968263
	15t16	14.570366	49.431980	36.266290	835.587643	54.070009
	17t18	18.335691	50.188234	15.538649	45.917943	45.612614
	19	4.728489	12.942764	4.007176	11.841522	11.762783
	20	931.529917	29.447854	10.273590	19.271625	19.486462
	21t22	90.326674	2821.000424	23.198240	363.916574	142.223056
	23	35.551012	77.346703	518.540713	250.877221	41.087803
	24	134.600162	414.014715	254.533734	1940.073909	858.289530
	25	36.346518	308.640250	35.017383	356.994995	269.405154
	26	40.469744	29.765072	8.536396	105.011698	39.029102
	27t28	307.140650	183.821644	39.748282	376.009535	192.166872
	29	22.279107	45.205836	36.726841	49.857619	28.242531
	30t33	13.228830	26.842204	21.807568	29.604327	16.769776
	34t35	8.080558	19.639293	18.150425	25.406816	11.405865
	36t37	110.342068	35.268883	21.252913	64.355459	29.135448
	E	143.282141	356.331393	124.990480	359.975571	187.401518
	F	109.473492	151.285346	935.469320	167.746506	93.523215
	50	230.146065	723.933947	115.779055	392.670136	231.061142
	51	276.108617	963.440976	976.941746	1308.409695	535.714326
	52	289.884901	858.592422	1003.348917	1302.437162	535.855098
	H	23.433113	167.690188	103.972792	168.583295	31.931233
	60	218.706857	510.934567	264.079532	529.984384	320.186820
	61	3.898249	15.919886	15.316702	13.456574	5.596796
	62	1.669552	11.293316	2.145138	6.468971	1.904794
	63	325.523642	634.162539	98.547353	630.174898	72.132960
	64	74.415816	321.615785	114.934853	143.609205	71.615890
	J	111.468536	441.552736	97.016295	296.043420	121.990028
	70	300.402024	541.707671	186.950544	98.494018	50.868687
	71t74	557.009305	2787.056493	1262.566488	2654.652613	1383.027258
...
RoW	20	59.254639	1.921591	0.293189	0.419081	0.896964
	21t22	10.939833	305.243836	2.718898	43.005354	12.827419
	23	25.409174	52.937556	426.332774	189.227887	25.663685
	24	45.776375	137.849053	88.781323	694.935609	304.297810
	25	4.080999	38.578026	4.433015	43.078751	33.263153
	26	1.540591	0.842093	0.991921	5.264341	1.845787
	27t28	19.202156	17.107983	2.899187	23.417741	40.398734
	29	4.226874	5.044632	4.710717	6.153871	1.448788
	30t33	0.581584	5.573272	0.601050	3.175119	2.472000
	34t35	0.295067	0.589052	0.144417	0.873163	0.518826

36t37	1.144750	1.872070	0.537275	2.212297	1.106440
E	0.173761	0.567528	20.428639	0.819155	0.253561
F	0.228649	0.150437	0.984623	0.227128	0.110467
50	0.012600	0.039960	0.049854	0.052195	0.025508
51	0.853480	4.639344	230.537056	6.007003	1.436657
52	0.267343	1.589181	0.516815	1.104595	0.320696
H	0.112285	0.844576	0.471689	0.894151	0.123268
60	0.435869	2.825101	234.376543	3.367219	0.414917
61	0.008164	0.032342	0.082525	0.046836	0.007787
62	1.580617	13.245788	2.369183	7.540798	2.183210
63	0.151014	0.529874	0.096219	0.398736	0.083757
64	0.065116	0.304461	0.102210	0.231570	0.101406
J	0.501054	3.041203	0.527920	1.128715	0.542113
70	3.690224	5.954991	2.422823	0.176277	0.184588
71t74	1.860358	9.415961	4.389323	8.278023	4.489652
L	0.248589	0.636937	0.238198	1.078893	0.160527
M	1.003295	4.474255	4.120725	4.517280	1.687657
N	1.749095	6.793279	0.164896	18.550374	0.174962
O	0.562692	2.533730	0.758305	2.141030	0.484539
P	0.000031	0.000096	0.000021	0.000116	0.000052

region	...	RoW			\
sector	...	63	64	J	
region sector	...				
AUS AtB	...	19.761917	0.001627	0.140044	
C	...	0.211888	0.034300	0.005817	
15t16	...	1.621756	1.588110	3.701685	
17t18	...	0.401032	0.181367	1.492579	
19	...	0.039708	0.017958	0.147786	
20	...	0.027123	0.082776	0.063378	
21t22	...	2.204172	10.102687	15.418063	
23	...	59.028995	4.487274	11.189084	
24	...	2.603711	1.444149	1.194342	
25	...	0.973279	2.427753	0.704863	
26	...	0.186875	0.421755	0.195426	
27t28	...	15.617744	7.959440	2.380652	
29	...	3.642359	30.309729	6.825963	
30t33	...	2.002236	16.661519	3.752291	
34t35	...	12.703247	8.769379	12.768384	
36t37	...	0.226041	1.816351	1.501001	
E	...	0.158169	0.222314	0.298537	
F	...	0.967638	3.020002	6.822620	
50	...	0.542576	1.055301	1.163450	
51	...	0.822974	0.880188	2.617088	
52	...	3.508417	8.412820	2.943161	
H	...	13.417060	5.803010	31.833836	
60	...	24.160506	27.963040	34.284311	
61	...	2.683645	27.565014	25.181994	

	62	...	56.890998	228.034250	187.103415
	63	...	23.523008	76.260841	59.942885
	64	...	0.183698	3.195211	4.168414
	J	...	10.691209	38.702169	87.947969
	70	...	1.371040	55.154968	51.222023
	71t74	...	13.609769	44.055788	110.840096
...
RoW	20	...	80.089178	38.342266	12.072937
	21t22	...	2356.505474	5432.580816	7910.918929
	23	...	940.236675	261.091828	154.913923
	24	...	259.001665	86.217475	90.520415
	25	...	1122.804886	762.399457	140.149436
	26	...	119.947154	60.337788	18.657645
	27t28	...	1298.244145	315.873918	171.199863
	29	...	697.828174	319.985813	314.995799
	30t33	...	818.546621	8308.403067	731.275786
	34t35	...	1252.400960	1279.511149	517.703475
	36t37	...	103.922288	113.850921	471.973179
	E	...	4605.709269	7342.310696	5072.978446
	F	...	982.318867	2242.910054	3428.713588
	50	...	522.931361	593.953802	533.712735
	51	...	3519.981195	7342.086914	2956.802019
	52	...	2200.648191	3472.751467	2002.609268
	H	...	5697.775736	2180.164612	7534.073416
	60	...	7703.649094	5140.042568	7039.531612
	61	...	319.845929	351.739712	393.717271
	62	...	532.908584	1341.365292	1312.401998
	63	...	3504.280492	1807.314685	1701.271137
	64	...	2172.194846	20440.020040	11710.122737
	J	...	7415.709712	8091.081325	38655.926218
	70	...	1727.970393	5199.410339	7493.075988
	71t74	...	6246.240339	12482.298686	24886.499828
	L	...	608.700335	709.004146	1595.531658
	M	...	91.678620	463.402398	1802.804212
	N	...	39.843405	106.735687	305.274672
	O	...	5414.944076	4371.240663	5815.525102
	P	...	1.324062	0.244611	1.104726

region					
sector		70	71t74	L	M
region sector					
AUS	AtB	0.043667	11.680006	3.113827	61.711687
	C	0.088101	14.418832	0.315809	0.182157
	15t16	2.743954	27.665274	14.865583	86.096798
	17t18	0.632427	1.492750	6.550554	0.878764
	19	0.062619	0.147803	0.648594	0.087010
	20	0.205954	0.790532	2.353193	2.554166
	21t22	6.195107	48.596887	56.890466	21.913868

	23	11.512628	36.303514	27.566443	17.222495
	24	9.237368	19.609616	28.268265	42.763999
	25	3.349751	6.259412	2.576908	3.199048
	26	0.657503	1.604297	1.160610	1.514350
	27t28	18.293034	25.094179	4.177664	5.533838
	29	28.606393	41.232405	7.775303	17.255305
	30t33	15.725181	22.665809	4.274151	9.485390
	34t35	12.610675	64.711016	21.845125	12.221815
	36t37	0.734132	5.457432	3.653690	3.040151
	E	1.804297	0.696923	0.819074	0.433625
	F	4.445031	6.978164	3.820432	4.213175
	50	0.619550	0.820108	3.190926	0.401245
	51	1.334045	8.044024	8.112138	13.165919
	52	4.183650	25.137596	9.350485	9.915846
	H	6.093826	44.016163	31.470631	22.222129
	60	29.296585	51.018691	66.987976	39.638572
	61	128.915922	66.003832	27.081238	6.700347
	62	40.923609	416.402802	518.481234	387.002863
	63	23.682097	132.257649	162.850890	120.792474
	64	0.398777	1.136361	4.488764	2.069959
	J	49.862781	73.172178	17.970300	40.594369
	70	81.507135	1.324506	14.642011	7.911327
	71t74	45.926864	97.594121	30.529568	59.538339
...	
RoW	20	26.345271	629.084892	1884.054328	2120.766430
	21t22	1128.083167	26948.165198	16081.684247	9758.327387
	23	129.072765	789.857491	574.234856	141.229352
	24	172.206457	1469.037784	568.857200	464.549470
	25	493.458933	3238.293742	384.585818	748.594202
	26	135.596128	389.433240	439.866913	502.917719
	27t28	733.140609	4511.622232	548.761711	543.615393
	29	181.223515	1068.322357	753.097670	271.294575
	30t33	808.302796	21372.986647	1858.083732	2393.918095
	34t35	418.842054	6459.762846	2627.353449	1120.680148
	36t37	181.271520	482.787859	773.675064	852.380832
	E	5141.941351	12517.586986	15224.240044	12110.158949
	F	7483.145885	5407.731676	10000.889751	6437.551020
	50	225.393607	984.574183	1437.815851	342.158934
	51	2056.102533	13468.319458	8755.578709	4942.954370
	52	1022.444732	6822.892202	4692.901833	2286.924606
	H	1565.207855	12160.455853	16699.714473	7750.382855
	60	1167.017249	10630.975028	10255.302827	4952.250191
	61	249.455177	1314.984794	916.026979	435.280134
	62	197.515757	2930.795161	3507.522443	2074.276308
	63	373.643784	3223.972374	4095.498850	1417.898957
	64	1468.996337	9246.282287	15881.469928	6012.447058
	J	5758.940639	19989.913847	23896.327061	9546.835763
	70	2705.769797	9495.163791	8103.153981	5036.228134

71t74	6015.478162	70478.453504	20430.836359	13305.638974
L	620.253937	1194.982705	4557.146576	518.488155
M	212.193133	1571.999239	7706.676330	5223.189539
N	70.364956	895.599603	1415.975389	840.530518
O	2897.014578	14958.970521	15443.391745	7897.357003
P	1.141934	50.552086	0.000000	1.089908

region				
sector		N	O	P
region sector				
AUS	AtB	9.898359	10.256983	0.000038
	C	0.273387	0.493510	0.000861
	15t16	46.736852	79.637133	0.006089
	17t18	2.252624	2.428735	0.001057
	19	0.223040	0.240478	0.000105
	20	1.827102	1.287573	0.010418
	21t22	17.326665	25.336693	0.025942
	23	18.549700	17.863948	0.011255
	24	112.870262	29.349604	0.052557
	25	4.867827	37.445945	0.008494
	26	1.381380	2.173674	0.001937
	27t28	4.012130	26.919745	0.049209
	29	6.985789	19.145209	0.031949
	30t33	3.840148	10.524287	0.017563
	34t35	5.288575	124.684698	0.005111
	36t37	2.261301	6.176947	0.330733
	E	0.588291	0.542892	0.000114
	F	4.269064	3.135682	0.006102
	50	0.715443	0.335334	0.000000
	51	10.882036	17.335205	0.125509
	52	14.368875	9.505814	0.045543
	H	15.193905	8.820562	0.024843
	60	6.561055	44.037984	0.014616
	61	2.041265	17.273710	0.000000
	62	28.960407	154.036071	0.000035
	63	9.267663	53.423142	0.000042
	64	1.476196	0.879504	0.000243
	J	12.824709	34.707057	0.069814
	70	0.554078	15.369971	0.000000
	71t74	18.124551	33.468160	0.097561
...	
RoW	20	1932.161785	914.755872	0.019687
	21t22	6011.875100	10934.428273	2.548514
	23	242.320646	423.896780	0.056465
	24	10243.419875	1465.864732	0.523719
	25	2193.842232	2876.932893	0.139278
	26	1116.624356	400.017883	0.111932
	27t28	931.955552	1147.905601	0.648180

29	3154.318917	314.337084	0.227289
30t33	2193.940291	4724.352592	1.743540
34t35	392.761307	7211.302026	0.984097
36t37	323.697939	504.572681	0.169314
E	12791.419341	10535.612431	4.831273
F	11354.319342	3839.537017	22.570339
50	1088.152585	555.415078	0.513696
51	19728.444632	9261.002711	4.393748
52	8058.010782	4256.950765	6.790453
H	5346.893566	4441.279950	1.070657
60	6210.736712	5982.700080	13.287645
61	803.452478	647.422802	0.257212
62	875.609110	1119.259998	0.346236
63	2264.941271	2108.901331	0.387366
64	7642.069297	4260.482612	8.457208
J	3563.218290	8195.767795	54.664765
70	6230.303094	7501.857446	0.007471
71t74	15560.647836	12040.071161	83.100716
L	921.621306	901.624286	0.000000
M	1488.980938	527.910083	0.001030
N	3513.234966	395.244668	0.000020
O	8895.570576	16565.733127	44.703532
P	4.570108	41.517464	3.460032

[1435 rows x 1435 columns]

WIOD includes several satellite accounts, which are stored as child objects in Pymrio. For example, in order to see the AIR emissions provided by WIOD:

In [7]: wiod.AIR.F

```
Out [7]: region          AUS
sector          AtB          C          15t16          17t18
stressor
CO2          6.471152e+03  2.331841e+04  3256.861259  392.819896
CH4          3.226169e+06  1.370016e+06  1221.450093  41.723574
N2O          6.527106e+04  1.243851e+02  527.652440  10.773378
NOX          2.000881e+05  1.709849e+05  70375.533177  3875.234721
SOX          1.976645e+04  4.713841e+04  45815.675397  1068.354291
CO          1.496859e+06  7.159254e+05  227663.413138  16225.875707
NMVOC        3.824729e+05  2.409498e+05  141642.740887  5460.933412
NH3          4.049434e+05  4.575323e+02  112.157985  4.313657

region
sector          19          20          21t22          23
stressor
CO2          91.570641  147.075293  2100.167306  7928.850694
CH4          6.112471  64.722688  189.787544  33785.867211
```

N2O	1.335362	14.793543	111.798406	146.523698
NOX	964.709338	9146.373832	36269.747108	18894.321469
SOX	265.958435	2521.542160	26680.445075	150018.069958
CO	4039.304699	38296.499606	82187.571692	56833.653485
NMVOC	1359.456610	12888.958231	28691.901753	65893.299291
NH3	0.449874	13.342974	48.333137	4.143371
region		...		RoW
sector	24	25	...	63
stressor			...	64
CO2	8832.607331	82.623337	...	4.530961e+04
CH4	768.018325	22.631731	...	2.031444e+04
N2O	10421.185919	6.723839	...	7.185897e+02
NOX	34546.808023	663.707765	...	1.419601e+05
SOX	91733.983039	182.976023	...	6.973313e+04
CO	393632.686253	2778.990301	...	1.385415e+06
NMVOC	105133.073315	935.288872	...	3.377663e+05
NH3	366.328954	4.740067	...	4.569925e+02
region				
sector	J	70	71t74	L
stressor				
CO2	17594.617416	11774.769870	5.787274e+04	1.118612e+05
CH4	3979.962245	5525.320218	1.617241e+04	8.865122e+04
N2O	342.725185	215.478088	1.228492e+03	2.995831e+03
NOX	68809.569199	60385.601411	2.027940e+05	4.224802e+05
SOX	33800.385037	29662.394375	9.961571e+04	2.075292e+05
CO	671525.284261	589314.228014	1.979104e+06	4.123062e+06
NMVOC	163718.896513	143675.714632	4.825085e+05	1.005209e+06
NH3	229.110878	244.663450	1.643215e+03	1.357458e+03
region				
sector	M	N	O	P
stressor				
CO2	25382.470248	4.575128e+04	4.851286e+04	0.0
CH4	5001.012041	1.394311e+04	1.360473e+07	0.0
N2O	272.108324	7.680525e+03	9.170715e+04	0.0
NOX	99579.016981	1.666314e+05	1.565510e+05	0.0
SOX	48914.840693	8.185206e+04	7.690040e+04	0.0
CO	971810.003503	1.626186e+06	1.527810e+06	0.0
NMVOC	236928.772636	3.964667e+05	3.724824e+05	0.0
NH3	113.829614	6.203676e+02	4.965187e+03	0.0

[8 rows x 1435 columns]

WIOD, however, does neither provide any normalized data (A-matrix, satellite account coefficient data) nor any consumption based accounts (footprints).

In order to calculate them, one could go through all the missing data and compute each ac-

count. Pymrio provides the required function, for example to calculate the A-matrix:

```
In [8]: x = pymrio.calc_x(Z=wiod.Z, Y=wiod.Y)
        A = pymrio.calc_A(Z=wiod.Z, x=x)
```

```
In [9]: A.head()
```

```
Out[9]: region          AUS
sector          AtB          C      15t16      17t18          19          20
region sector
AUS   AtB      0.095452  0.000346  0.220811  0.086780  0.096757  0.093637
      C        0.000350  0.031718  0.002684  0.001828  0.002039  0.001426
      15t16    0.022535  0.000831  0.095444  0.011052  0.012323  0.001456
      17t18    0.000793  0.000362  0.001540  0.057481  0.064090  0.001833
      19       0.000204  0.000093  0.000397  0.014824  0.016528  0.000473
```

```
region
sector          21t22          23          24          25          ...
region sector
AUS   AtB      0.009559  0.000000  0.008643  0.008967          ...
      C        0.002035  0.220910  0.007746  0.002002          ...
      15t16    0.001731  0.001811  0.033481  0.005163          ...
      17t18    0.001757  0.000776  0.001840  0.004355          ...
      19       0.000453  0.000200  0.000474  0.001123          ...
```

```
region          RoW
sector          63          64          J          70
region sector
AUS   AtB      1.143737e-04  5.153717e-09  2.285107e-07  7.786856e-08
      C        1.226316e-06  1.086802e-07  9.492337e-09  1.571046e-07
      15t16    9.386042e-06  5.031958e-06  6.040074e-06  4.893121e-06
      17t18    2.321002e-06  5.746660e-07  2.435455e-06  1.127767e-06
      19       2.298108e-07  5.689976e-08  2.411432e-07  1.116643e-07
```

```
region
sector          71t74          L          M          N
region sector
AUS   AtB      1.517644e-05  4.463792e-06  1.377724e-04  2.552552e-05
      C        1.873514e-05  4.527246e-07  4.066696e-07  7.050016e-07
      15t16    3.594692e-05  2.131039e-05  1.922126e-04  1.205233e-04
      17t18    1.939607e-06  9.390472e-06  1.961856e-06  5.808983e-06
      19       1.920475e-07  9.297846e-07  1.942504e-07  5.751684e-07
```

```
region
sector          0          P
region sector
AUS   AtB      2.604967e-05  6.252650e-09
      C        1.253368e-06  1.432423e-07
```

```

15t16    2.022545e-04  1.013494e-06
17t18    6.168260e-06  1.759452e-07
19       6.107418e-07  1.742097e-08

```

[5 rows x 1435 columns]

Alternatively, Pymrio provides a function which iterates through all missing accounts and calculates them:

```
In [10]: wiod.calc_all()
```

```
Out[10]: <pymrio.core.mriosystem.IOSystem at 0x7f76670548d0>
```

At this point, a basic EE MRIO analysis is accomplished. For example, the regional consumption based accounts of the AIR emissions are now given by:

```
In [11]: wiod.AIR.D_cba_reg
```

```

Out[11]: region          AUS          AUT          BEL          BGR  \
stressor
CO2      4.404070e+05  1.022100e+05  1.586176e+05  42924.986975
CH4      4.275465e+06  7.599975e+05  1.030354e+06  464018.748607
N2O      9.588178e+04  3.086814e+04  4.609171e+04  13203.713081
NOX      2.359815e+06  3.324339e+05  4.508892e+05  142917.818720
SOX      2.399335e+06  1.983047e+05  3.702525e+05  400357.951750
CO       2.173900e+07  1.371366e+06  2.167114e+06  703172.284772
NMVOC    3.101630e+06  3.582680e+05  5.920832e+05  190582.650539
NH3      3.851776e+05  9.254548e+04  1.245648e+05  45897.394639

region          BRA          CAN          CHN          CYP  \
stressor
CO2      4.059629e+05  5.659664e+05  5.031700e+06  13943.187686
CH4      1.352464e+07  4.068558e+06  5.433871e+07  157009.091900
N2O      5.899229e+05  1.634371e+05  1.831795e+06  3200.309665
NOX      2.786076e+06  1.904551e+06  1.925370e+07  35972.513098
SOX      1.699074e+06  2.088103e+06  3.245490e+07  43500.967386
CO       2.681292e+07  7.525147e+06  9.904520e+07  144686.477852
NMVOC    5.323333e+06  2.131757e+06  2.016103e+07  43518.270216
NH3      1.345046e+06  4.204562e+05  6.415339e+06  8931.594745

region          CZE          DEU          ...          PRT  \
stressor
CO2      108758.745642  1.054136e+06  ...          7.658922e+04
CH4      780222.089424  6.668537e+06  ...          8.948877e+05
N2O      27441.728571  2.914646e+05  ...          3.091511e+04
NOX      292246.717821  2.701648e+06  ...          3.025544e+05
SOX      225907.785277  1.951840e+06  ...          1.918064e+05
CO       829881.093571  1.099191e+07  ...          1.194956e+06
NMVOC    273912.655126  2.923060e+06  ...          4.003833e+05

```

NH3	67224.140441	8.505438e+05	...	8.987684e+04
-----	--------------	--------------	-----	--------------

region	ROU	RUS	SVK	SVN	\
stressor					
CO2	1.173831e+05	1.311461e+06	40459.233377	24251.728341	
CH4	1.344168e+06	1.532052e+07	411099.098085	182504.004554	
N2O	5.163863e+04	4.422776e+05	13658.674182	7011.741463	
NOX	3.380263e+05	4.444685e+06	125333.624730	73379.182597	
SOX	4.979996e+05	1.398364e+06	103186.610826	48180.228365	
CO	2.055640e+06	2.165403e+07	455232.910264	580264.681074	
NMVOC	6.065405e+05	4.179851e+06	138654.031320	142881.841642	
NH3	1.911781e+05	7.889417e+05	35515.968170	23476.633339	

region	SWE	TUR	TWN	USA	RoW
stressor					
CO2	9.434506e+04	3.494179e+05	2.246294e+05	6.210161e+06	5.620778e+06
CH4	7.352664e+05	3.652537e+06	1.104729e+06	3.917121e+07	7.560548e+07
N2O	3.576881e+04	9.538255e+04	3.551477e+04	1.182906e+06	3.590470e+06
NOX	3.524920e+05	1.797639e+06	8.632669e+05	1.845556e+07	3.504645e+07
SOX	2.078760e+05	1.548830e+06	1.075249e+06	1.523013e+07	2.860410e+07
CO	2.048666e+06	4.860666e+06	4.927789e+06	1.005814e+08	3.566157e+08
NMVOC	5.666544e+05	1.738178e+06	1.095519e+06	2.095710e+07	4.208942e+07
NH3	8.488958e+04	5.979386e+05	1.032068e+05	3.090159e+06	8.572543e+06

[8 rows x 41 columns]

In [12]: wiod.AIR.unit

```
Out[12]: unit      unit
stressor
CO2         Gg
CH4         t
N2O         t
NOX         undef
SOX         undef
CO          t
NMVOC       t
NH3         t
```

Pymrio can be linked with the [country converter coco](#) to ease the aggregation of MRIO and results into different classifications. Using the country converter, WIOD can be aggregated into EU and non-EU countries with singling out Germany by:

```
In [13]: import country_converter as coco
wiod.aggregate(region_agg = coco.agg_conc(original_countries='WIOD',
                                           aggregates=[{'DEU': 'DEU', 'GBR': 'GBR'}, 'EU'],
                                           missing_countries='Other',
                                           merge_multiple_string=None))
```

```
Out [13]: <pymrio.core.mriosystem.IOSystem at 0x7f76670548d0>
```

We rename the EU account to reflect that it does not include Germany:

```
In [14]: wiod.rename_regions({'EU': 'Rest of EU'})
```

```
Out [14]: <pymrio.core.mriosystem.IOSystem at 0x7f76670548d0>
```

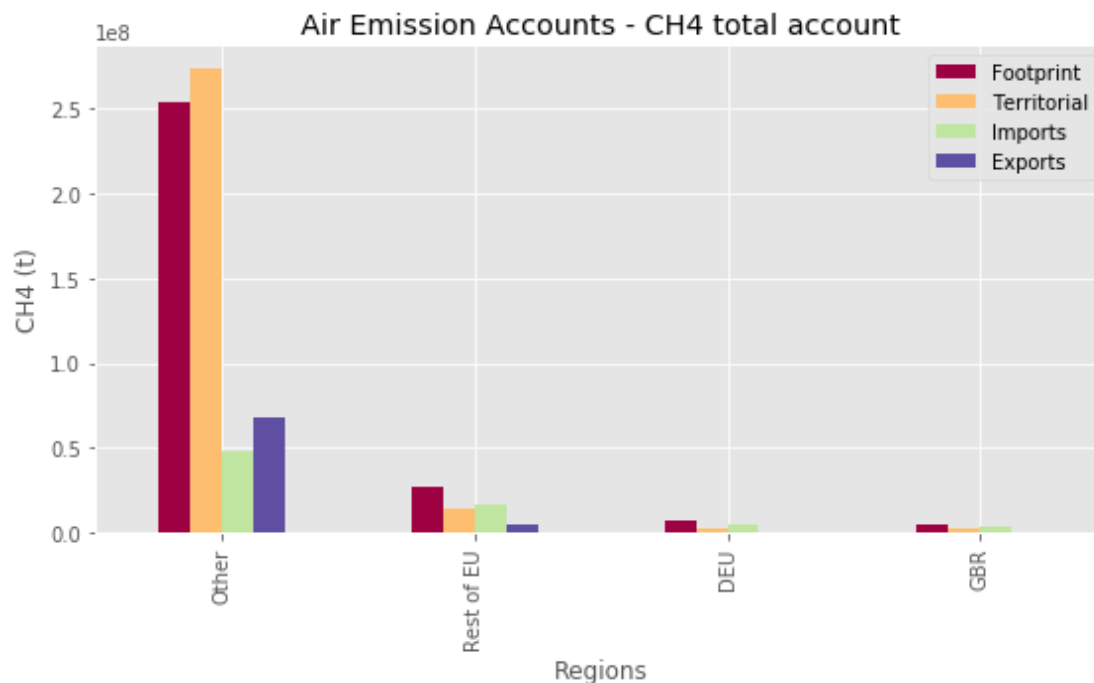
The regional footprint account are now:

```
In [15]: wiod.AIR.D_cba_reg
```

```
Out [15]: region          Other    Rest of EU          DEU          GBR
stressor
CO2      2.436179e+07  3.472823e+06  1.054136e+06  7.397044e+05
CH4      2.540661e+08  2.711250e+07  6.668537e+06  5.235498e+06
N2O      9.705186e+06  1.128531e+06  2.914646e+05  2.118832e+05
NOX      1.043111e+08  1.093267e+07  2.701648e+06  2.164933e+06
SOX      1.037493e+08  8.344435e+06  1.951840e+06  1.421854e+06
CO       7.661455e+08  5.466639e+07  1.099191e+07  1.068169e+07
NMVOC    1.280392e+08  1.577316e+07  2.923060e+06  2.986943e+06
NH3      2.672782e+07  3.493227e+06  8.505438e+05  5.900984e+05
```

To visualize for example the CH4 accounts:

```
In [16]: import matplotlib.pyplot as plt
with plt.style.context('ggplot'):
    wiod.AIR.plot_account('CH4', figsize=(8,5))
    plt.savefig('/tmp/wiod/airch4.png', dpi=300)
    plt.show()
```



To calculate the source (in terms of regions and sectors) of a certain stressor or impact driven by consumption, one needs to diagonalize this stressor/impact. This can be done with Pymrio by:

```
In [17]: diag_CH4 = wiod.AIR.diag_stressor('CH4')
```

and be reassigned to the aggregated WIOD system:

```
In [18]: wiod.CH4_source = diag_CH4
```

In the next step the automatic calculation routine of Pymrio is called again to compute the missing accounts in this new extension: and be reassigned to the aggregated WIOD system:

```
In [19]: wiod.calc_all()
```

```
Out[19]: <pymrio.core.mriosystem.IOSystem at 0x7f76670548d0>
```

The diagonalized CH4 data now shows the source and destination of the specified stressor (CH4):

```
In [20]: wiod.CH4_source.D_cba.head()
```

```
Out[20]: region          Other          \
          sector          AtB          C          15t16          17t18
region sector
Other AtB  6.120041e+07  8.455234e+04  3.658411e+07  2.988418e+06
          C  1.008359e+06  6.714292e+06  2.047332e+06  6.420307e+05
          15t16  3.968202e+03  6.218228e+01  8.736127e+04  5.178453e+02
          17t18  9.369869e+01  2.287802e+01  2.464623e+02  1.185856e+04
          19  5.587156e+00  1.040420e+00  1.268140e+01  1.252185e+02

region          19          20          21t22          23          \
sector
region sector
Other AtB  867172.684194  230646.602537  441913.627992  1.766666e+05
          C  117665.005135  33160.714629  281538.575874  5.643346e+06
          15t16  639.802755  24.298048  128.945944  1.099366e+02
          17t18  200.126039  4.500219  68.017013  3.330082e+01
          19  1231.987687  0.344568  4.205279  1.580128e+00

region          ...          GBR          \
sector          24          25          ...          63
region sector
Other AtB  6.045454e+05  197135.975631  ...  669.903691
          C  1.564758e+06  231675.430540  ...  1405.758158
          15t16  4.216024e+02  83.091425  ...  0.460965
          17t18  7.862589e+01  59.680321  ...  0.093075
          19  4.194052e+00  3.043624  ...  0.008632
```

region					
sector		64	J	70	71t74
region	sector				
Other	AtB	3406.973739	8805.678036	6656.233633	4123.121986
	C	9357.742729	20287.540899	18680.219527	8586.783042
	15t16	3.215184	8.818689	5.805941	3.871073
	17t18	0.848278	1.738428	1.444981	0.919367
	19	0.117312	0.225215	0.171093	0.106037

region					
sector		L	M	N	0
region	sector				
Other	AtB	19984.727842	9398.054711	44359.672517	12305.895420
	C	49408.298618	15853.401016	93562.146555	23512.720979
	15t16	18.671491	8.695184	44.370659	11.707402
	17t18	6.608830	2.031215	12.658218	3.765588
	19	1.104701	0.343349	1.430827	0.646012

region		
sector		P
region	sector	
Other	AtB	3.055303
	C	4.041194
	15t16	0.001748
	17t18	0.001256
	19	0.000075

[5 rows x 140 columns]

In this square footprint matrix, every column represents the amount of stressor occurring in each region - sector driven by the consumption stated in the column header. Conversely, each row states where the stressor impacts occurring in the row are distributed due (from where they are driven).

```
In [21]: CH4_source_reg = wiod.CH4_source.D_cba.groupby(
        level='region', axis=0).sum().groupby(
        level='region', axis=1).sum()
```

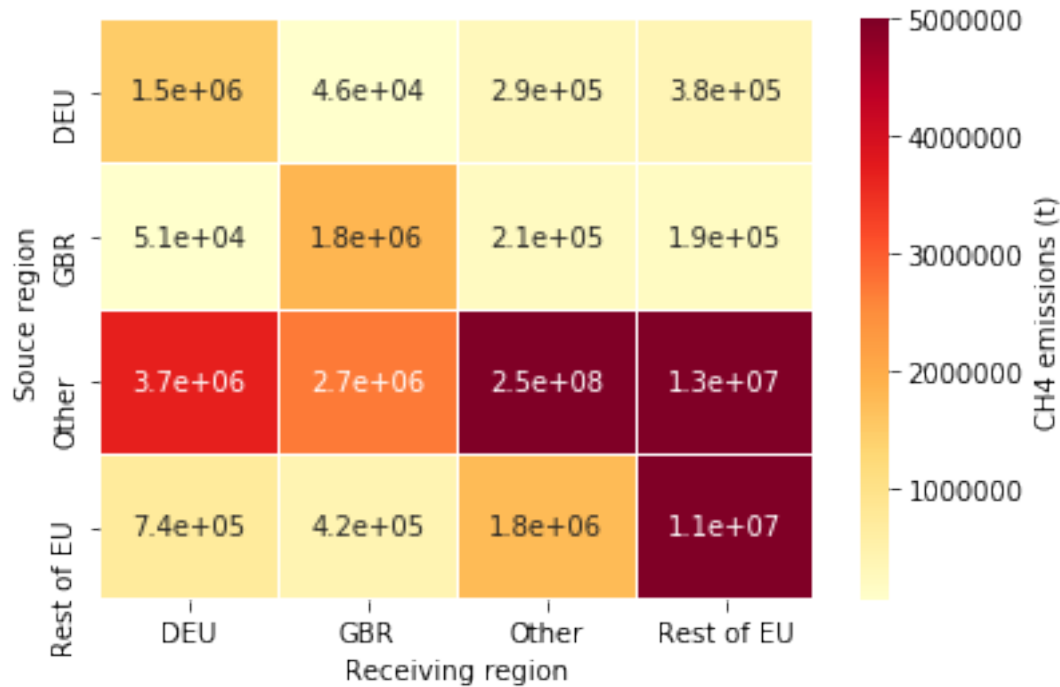
```
In [22]: CH4_source_reg
```

region		DEU	GBR	Other	Rest of EU
region					
DEU		1.485343e+06	4.634238e+04	2.892830e+05	3.819713e+05
GBR		5.139252e+04	1.833541e+06	2.112405e+05	1.879226e+05
Other		3.696832e+06	2.711860e+06	2.457410e+08	1.317725e+07
Rest of EU		7.402886e+05	4.186665e+05	1.756700e+06	1.128755e+07

```
In [23]: import seaborn as sns
        CH4_source_reg.columns.name = 'Receiving region'
```



```
CH4_source_reg.index.name = 'Souce region'
sns.heatmap(CH4_source_reg, vmax=5E6,
            annot=True, cmap='YlOrRd', linewidths=0.1,
            cbar_kws={'label': 'CH4 emissions ({})'.format(wiod.CH4_source.unit.unit[0])})
plt.savefig('/tmp/wiod/airch4_source_reg.png', dpi=300)
plt.show()
```



Storing the MRIO database can be done with

```
In [24]: storage_path = '/tmp/wiod/aly'
wiod.save_all(storage_path)
```

```
Out[24]: <pymrio.core.mriosystem.IOSystem at 0x7f76670548d0>
```

From where it can be received subsequently by:

```
In [25]: wiod = pymrio.load_all(storage_path)
```

The meta attribute of Pymrio mentioned at the beginning kept track of all modifications of the system. This can be shown with:

```
In [26]: wiod.meta
```

```
Out[26]: Description: WIOD metadata file for pymrio
MRIO Name: WIOD
System: industry-by-industry
```

Version: data13

File: /tmp/wiod/aly/metadata.json

History:

```
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/factor_input
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/SEA
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/AIR
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/CO2
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/EM
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/EU
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/lan
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/mat
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/wat
20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/CH4_source
... (more lines in history)
```

Custom notes can be added to the meta with:

```
In [27]: wiod.meta.note("Custom note")
```

The history of the meta data can be filtered for specific entries like:

```
In [28]: wiod.meta.file_io_history
```

```
Out[28]: ['20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/factor_inpu
'20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/SEA',
'20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/AIR',
'20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/CO2',
'20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/EM',
'20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/EU',
'20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/lan',
'20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/mat',
'20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/wat',
'20180117 09:38:37 - FILEIO - Added satellite account from /tmp/wiod/aly/CH4_source
'20180117 09:38:37 - FILEIO - Loaded IO system from /tmp/wiod/aly',
'20180117 09:38:36 - FILEIO - Saved WIOD to /tmp/wiod/aly',
'20180117 09:38:32 - FILEIO - Extension wat parsed from /tmp/wiod/raw',
'20180117 09:38:31 - FILEIO - Extension mat parsed from /tmp/wiod/raw',
'20180117 09:38:30 - FILEIO - Extension lan parsed from /tmp/wiod/raw',
'20180117 09:38:30 - FILEIO - Extension EU parsed from /tmp/wiod/raw',
'20180117 09:38:28 - FILEIO - Extension EM parsed from /tmp/wiod/raw',
'20180117 09:38:26 - FILEIO - Extension CO2 parsed from /tmp/wiod/raw',
'20180117 09:38:25 - FILEIO - Extension AIR parsed from /tmp/wiod/raw',
'20180117 09:38:23 - FILEIO - SEA file extension parsed from /tmp/wiod/raw',
'20180117 09:38:17 - FILEIO - WIOD data parsed from /tmp/wiod/raw/wiot08_row_sep12.
'20180112 09:23:36 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/water
'20180112 09:23:36 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/mater
'20180112 09:23:35 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/land
'20180112 09:23:35 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/AIR/
'20180112 09:23:34 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/CO2/
```

```
'20180112 09:23:33 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/EM/EM/
'20180112 09:23:32 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/EU/EU/
'20180112 09:23:32 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/SEA/SEA/
'20180112 09:23:31 - FILEIO - Downloaded http://www.wiod.org/protected3/data13/updat
```

This tutorial gave a short overview about the basic functionality of Pymrio. For more information about the capabilities of pymrio check the [online documentation](#).