NDCmitiQ

Release 1.0

Annika Guenther

Jul 22, 2020

CONTENTS:

Comments: in the code, use (3 times ” newline text newline 3 times “) for comments. Do not forget the new lines, else it will not appear in this documentation.

This documentation of the main functions to preprocess data and quantify mitigation targets from within the NDCs (tool NDCmitiQ), includes information retrieved from the different py-files of the tool. It does not include information from all py-files in this repository.

This documentation shall be seen as an add-on to the information given in the manuscript describing the methodology behind NDCmitiQ. As soon as the paper is published, its DOI will be added here.

The required pandas packages can be found in requirements.txt, and information on how to run the code can be found in the README in the main folder.

CHAPTER ONE

SETUP_METADATA

setup_metadata()

Set up the general metadata. If you want/need to change the folder for the preprocessed data, do it here.

Returns:

meta [class.] Meta data needed in the quantifications and plotting routines.

meta.path: paths to different folders.

meta.isos: iso3 codes of EARTH, EU, conversion from iso3 to short country names.

SSPs: Shared Socioeconomic Pathways; info on available marker scenarios.

Nomenclature: table meta-data nomenclature.

PRIMAP-hist: info on sources and scenarios (emi, pop, gdp).

Default-units

GWPs

LULUCF source-priorisation

Gases: basket-members, labels.

Sectors: main sectors, labels.

Categories: main categories, labels.

Nice labels for sources

NDC types.

CHAPTER TWO

PREPROCESSING_GENERAL

General preprocessing (covered part of emissions in preprocessing_current_pccov.py).

Fill missing data, SSPs (emi, pop, gdp), LULUCF emissions.

Load and write out information retrieved from NDCs to 'ndcs_info.csv' into folder meta.path.preprocess. Includes information on Parties' target types, years, conditionality, range, coverage.

Read in all datatables in folder matlab_tables.

All tables are read into the class 'database', with the tablenames as main-attributes. The table meta-data are classes again, with attributes 'entity', 'category', 'data', 'unit', etc. (see meta.nomenclature.attrs).

As we only use one GWP, replace the AR4 strings by '' (e.g., KYOTOGHGAR4_IPCM0EL_TOTAL_NET_HISTORY_CRF2019.csv is read in with tablename 'KYOTOGHG_IPCM0EL_TOTAL_NET_HISTORY_CRF2019').

SSPs: check if there are countries that do have data in some SSPs but not in others and use the average over the available SSPs for the non-available SSPs. For countries that have PRIMAP-hist data but no SSP data at all (for marker scenarios of SSP1 to SSP5), use the PRIMAP-hist data and use the linear regression over the last 6 available years as future estimates. Only happens for countries with very low emissions.

The SSPs only have data for FGASES, not separated into HFCS, PFCS, SF6 and NF3. For the calculation of the future pc_cov, a split into the four subgroups is performed. The historical share per gas/basket is kept constant and applied to the future FGASES-basket. The mean over the last 6 available years is used.

Check if the SSPs (KYOTOGHG_IPCM0EL per SSP scenario) are consistent with the given PRIMAP-hist data.

Prepare LULUCF data. Problems with LULUCF data: high inter-annual variability, negative / positive emissions, and the data we use are not consistent with the time series used in the pathway extension.

Make one LULUCF table, with the chosen KYOTOGHG_IPCMLULUCF time series. Prioritisation of data-sources as given in meta.lulucf.source_prioritisation. When a source has data, use them, and fill data gaps with 'constant filling'.

Interpolation & forward extrapolation: last value kept constant. Backward extrapolation: first available value after 1990 used for 1990 to year of first available value).

Other option: linear interpolation, but constant extrapolation (with mean over several years).

If KYOTOGHG is calculated here as sum over CO2 + CH4 + N2O: sum up the already inter- & extrapolated time series.

Additionally to prioritising CRF, BRU, UNFCCC, FAO in that order, put FAO or UNFCCC to the first place in the LULUCF prioritisation, for comparison runs.

Create NDC exclLU pathways. Use PRIMAP-hist v2.1 HISTCR up to 2017 and then use given ndcs_emi_exclLU when available, with linear interpolation between values.

For the countries without exclLU values check if they have inclLU values. If so use the onlyLU data from the NDC (or other options, as used in the NDC quantifications) to derive onlyLU emissions. Use the given inclLU values and the corresponding onlyLU values to derive exclLU, and then interpolate linearly between them (again, with PRIMAP-hist for KYOTOGHG_IPCM0EL up to 2017).

EMI onlyLU.

If there are onlyLU values available for 2010 or later, use them for constant extrapolation (use their average) if no future values are available.

EMI exclLU.

Check if there is exclLU data available from the NDC for years > 2017. If so, use them with linear interpolation from PRIMAP-hist to the first year, and inbetween. After last year: use SSP growth rates.

- If ndc_exclLU available: use ndc_exclLU.
- If ndc_inclLU available: use ndc_inclLU - onlyLU.
- Else: use external_exclLU.

Put in PRIMAP-hist data to 1990 - 2017. And interpolate linearly. Extrapolate with the SSP growth rates.

CHAPTER

THREE

PREPROCESSING.PREP_READ_IN_TABLES

prep_read_in_tables(file, path_to_folder, database, meta)

Read in all datatables in folder.

All read into class 'database', with the tablenames as attributes. Replace AR4 strings by '' (e.g., KYOTOGHGAR4_IPCM0EL_TOTAL_NET_HISTORY_CRF2019.csv is read in with tablename 'KYOTOGHG_IPCM0EL_TOTAL_NET_HISTORY_CRF2019'). The attributes are classes again, with attributes entity, category, data, family, etc. (see meta.nomenclature.attrs).

CHAPTER

FOUR

PREPROCESSING.PREP_SSPS_FILL_GAPS

prep_ssps_fill_gaps(database, info_per_country, meta, nrvalues)

SSPs: check if there are countries that do have data in some SSPs but not in others and use the average over the available SSPs for the non-available SSPs. For countries that have PRIMAP-hist data but no SSP data at all (for SSP1 to SSP5), use the PRIMAP-hist data and use the linear regression over the last 6 available years as future estimates. Only happens for countries with very low emissions.

fill_values(database, info_per_country, info_act, primap_extrapol, nrvalues, ent, ssp, ssps_test, ssps_mean)

For countries for which there are data available for some SSPs, but not for others, use the average over the available SSPs for the non-available SSPs. And for countries that have PRIMAP-hist data, but no SSP data at all, use the PRIMAP-hist data and use the linear regression over the last 6 available years as future estimates.

CHAPTER

FIVE

PREPROCESSING.PREP_SSPS_SPLIT_FGASES

prep_ssps_split_fgases(meta, database, nrvalues)

The SSPs only have data for FGASES, not separated into HFCS, PFCS, SF6 and NF3. For the calculation of the future pc_cov, a split into the four subgroups is performed. The historical share per gas/basket is kept constant and applied to the future FGASES-basket. The mean over the last 6 available years is used.

calc_and_store_data(database, meta, fgases_share, nrvalues)

CHAPTER

SIX

PREPROCESSING.PREP_LULUCF

prep_lulucf(database, meta, prios, srce_name, info_per_country, nrvalues, interpolation_method)

Prepare LULUCF data.

Make one LULUCF table, with the chosen KYOTOGHG_IPCMLULUCF time series. Prioritisation of data-sources as given in prios. When a source has data, use them, and fill data gaps with 'constant filling' (interpolation & forward extrapolation: mean over last values kept constant, backward extrapolation: mean over first available values kept constant). If KYOTOGHG is calculated here from CO2 + CH4 + N2O: sum up the already inter- & extrapolated time series.

Problems with LULUCF data: high inter-annual variability, negative / positive emissions, and the data we use are not consistent with the time series used in the pathway extension.

interpolation_method: 'constant' or 'linear'.

calc_data()

For the future, use the mean over 2010 to whatever the most recent historical value is, or if no values are available starting from 2010, use the last value. For the backward extrapolation use the mean over 1990 to 1997, or the first available value.

store_data()

Only use a source if at least 6 values are available for 1990 - 2017. If no other source has data, then use a source with less than 6 values available nevertheless. Store the DataFrames with various sources combined to one datatable in lulucf_table.

Only use a source if at least 6 values are available for 1990 - 2017. If no other source has data, then use a source with less than 6 values available nevertheless. Store the DataFrames with various sources combined to one datatable in lulucf_table.

CHAPTER

SEVEN

PREPROCESSING_CURRENT_PC_COV

Preprocessing to calculate the part of emissions covered by national mitigation targets.

Read in all datatables in folder meta.path.preprocess/tables.

All read into class 'database', with the tablenames as attributes. The meta-data are classes again, with attributes entity, category, data, unit, etc. (see meta.nomenclature.attrs).

Calculate the part of historical emissions that is covered by an NDC.

If the country does not have an (I)NDC: nothing is covered.

Else:

- Assessment based on PRIMAP-hist HISTCR emissions time series.
- Categories and gases assessed (per country):
  - Main categories (IPC1, IPC2, IPCMAG, IPC4 and IPC5; namely Energy, IPPU, Agriculture, Waste and Other; excludes LULUCF).
  - Kyoto GHGs: CO2, CH4, N2O, HFCS, PFCS, SF6, NF3.
- For each of these categories / gases, the information on whether they are covered by the country's NDC is provided (csv-input, assessed by A. Günther).
- If no information is available for all gases: CO2, CH4, and N2O are assumed to be covered (in the csv-file already).
- If any of HFCS, PFCS, SF6 or NF3 is covered, put IPPU to covered (F-gases only relevant in IPC2).
- If all sectors (IPC1, 2, MAG, 4) are covered, the category "Other" (IPC5) is set to "YES" (in the csv-file already).
- For all category + gas combinations, the emissions are counted as covered, if neither the category nor the gas are assumed not to be covered (neither category nor gas can contain a "NO").

Here, matrices on the coverage (Yes: covered, No: not-covered) are created.

We do not include the information on the EU28, but put the information into each of the member-states.

You can chose to setup your own coverage.used_per_gas_per_sec. If you do so, write out a file to say what you have chosen!!!

E.g. put all Energy and CO2 to covered ('YES'), and the rest to 'NO'. Or: all ANNEX-I parties cover everything, and the rest only Energy and CO2.

The covered part of emissions (in 'emissions') is calculated here for historical years, for which data per sector and gas combination are available. The assessment is based on the information in coverage.used_per_combi (all entries are already 'YES' or 'NO', no 'NAN' entries). All combinations with 'YES' are summed up to emicov_his, and all combinations with 'NO' are summed up to eminotcov_his.

For KYOTOGHG_IPCM0EL, excluding LULUCF.

Additionally calculate the historical emissions not-/covered for perGas_IPCM0EL and KYOTOGHG_perCategory. These values are not needed in further calculations, but nice to have.

Calculate the part of future emissions covered by an NDC (KYOTOGHG_IPCM0EL)

- For countries that cover everything: set pccov_fut to 1 (100%).
- For countries that cover nothing: set pccov_fut to 0 (0%).
- **For countries that cover all sectors (excl. LULUCF), but not all gases:** the SSP entity_IPCM0EL emissions per gas are used to calculate pc_cov_fut.
    - SSP data are available for KYOTOGHG, CO2, CH4, N2O and FGASES:
        - For countries that cover only some FGASES, the % share between HFCS, PFCS, SF6 and NF3 is kept constant (at mean over last 6 available PRIMAP-hist values).
        - The share per gas is applied to the future KYOTOGHG_IPCM0EL emissions data.
- For countries that do not cover all sectors:
    - Calculate the slope of pc_cov_his (2010 to most recent year with available data (“mry”)).
        - If abs(slope) < lim_slope: use the mean over 2010 to mry.
        - If abs(slope) > lim_slope: calculate pc_cov_fut from the correlation between emi_tot_his and emi_cov_his. For 2010 to mry.
            - If any(pc_cov_fut) > 90%, but not all(pc_cov_fut) > 90% –> set the pc_cov_fut > 90% to 90%.
            - If any(pc_cov_fut) < 10%, but not all(pc_cov_fut) < 10% –> set the pc_cov_fut < 10% to 10%.
            - If any(pc_cov_fut) > 100% or < 0% use the mean instead.
- If no future emissions data are available: use the mean over 2010 to mry.

The future emicov / pccov values depend on the chosen SSP scenario.

One can give a preference for the calculation method of pccov_fut. preference_pccov_fut can be ‘mean’ or ‘corr’.

‘mean’: Check for the countries for which the slope of a regression to the last available years of pccov_his is less than slope_lim. Use the mean over the years as pccov_fut. For the others check the correlation between emitot and emicov and decide whether to better use this correlation for pccov_fut.

‘corr’: Take the correlation between emitot and emicov, unless it is too ‘bad’, then take the mean.

Default: ‘corr’.

Update the current pc_cov-folder in setup_metadata after running preprocessing_current_pc_cov.py.

CHAPTER

EIGHT

PREPROCESSING.PREP_COVERAGE

prep_coverage(meta, infos_from_ndcs, info_per_country)

Calculate the part of historical emissions that is covered by an NDC.

If the country does not have an (I)NDC: nothing is covered.

Else:

- Assessment based on PRIMAP-hist HISTCR emissions time series.
- Categories and gases assessed (per country):
  - Main categories (IPC1, IPC2, IPCMAG, IPC4 and IPC5; namely Energy, IPPU, Agriculture, Waste and Other; excludes LULUCF).
  - Kyoto GHGs: CO2, CH4, N2O, HFCS, PFCS, SF6, NF3.
- For each of these categories / gases, the information on whether they are covered by the country's NDC is provided (csv-input, assessed by A. Günther).
- If no information is available for all gases: CO2, CH4, and N2O are assumed to be covered (in the csv-file already).
- If any of HFCS, PFCS, SF6 or NF3 is covered, put IPPU to covered (F-gases only relevant in IPC2).
- If all sectors (IPC1, 2, MAG, 4) are covered, the category "Other" (IPC5) is set to "YES" (in the csv-file already).
- For all category + gas combinations, the emissions are counted as covered, if neither the category nor the gas are assumed not to be covered (neither category nor gas can contain a "NO").

Here, matrices on the coverage (Yes: covered, No: not-covered) are created.

We do not include the information on the EU28, but put the information into each of the member-states.

current_coverage(meta, iso3, coverage, info_per_country)

CHAPTER

NINE

PREPROCESSING.PREP_COVERED_EMISSIONS_HIS

prep_covered_emissions_his(database, coverage, meta, primap)

The covered part of emissions (in ‘emissions’) is calculated here for historical years, for which data per sector and gas combination are available. The assessment is based on the information in coverage.used_per_combi (all entries are ‘YES’ or ‘NO’). All combinations with ‘YES’ are summed up to emicov_his, and all combinations with ‘NO’ are summed up to eminotcov_his.

For KYOTOGHG_IPCM0EL, excluding LULUCF.

testing(new_kyoto_ipcm0el, database, meta)

Test if the sum over the covered and not-covered emissions (KYOTOGHG_IPCM0EL) sum up to the original KYOTOGHG_IPCM0EL in ‘database’.

CHAPTER

TEN

PREPROCESSING.PREP_COVERED_EMISSIONS_FUT

prep_covered_emissions_fut(database, meta, coverage, info_per_country, preference_pccov_fut, primap, first_year_for_slope, slope_lim, rvalue_lim)

Calculate the part of future emissions covered by an NDC (KYOTOGHG_IPCM0EL)

- For countries that cover everything: set pccov_fut to 1 (100%).
- For countries that cover nothing: set pccov_fut to 0 (0%).
- **For countries that cover all sectors (excl. LULUCF), but not all gases:** the SSP entity_IPCM0EL emissions per gas are used to calculate pc_cov_fut.
  - SSP data are available for KYOTOGHG, CO2, CH4, N2O and FGASES:
    - For countries that cover only some FGASES, the % share between HFCS, PFCS, SF6 and NF3 is kept constant (at mean over last 6 available PRIMAP-hist values).
    - The share per gas is applied to the future KYOTOGHG_IPCM0EL emissions data.
- For countries that do not cover all sectors:
  - Calculate the slope of pc_cov_his (2010 to most recent year with available data (“mry”)).
    - If abs(slope) < lim_slope: use the mean over 2010 to mry.
    - If abs(slope) > lim_slope: calculate pc_cov_fut from the correlation between emi_tot_his and emi_cov_his. For 2010 to mry.
      - If any(pc_cov_fut) > 90%, but not all(pc_cov_fut) > 90% –> set the pc_cov_fut > 90% to 90%.
      - If any(pc_cov_fut) < 10%, but not all(pc_cov_fut) < 10% –> set the pc_cov_fut < 10% to 10%.
      - If any(pc_cov_fut) > 100% or < 0% use the mean instead.
- If no future emissions data are available: use the mean over 2010 to mry.

The future emicov / pccov values depend on the chosen SSP scenario.

One can give a preference for the calculation method of pccov_fut. preference_pccov_fut can be ‘mean’ or ‘corr’.

‘mean’: Check for the countries for which the slope of a regression to the last available years of pccov_his is less than slope_lim. Use the mean over the years as pccov_fut. For the others check the correlation between emitot and emicov and decide whether to better use this correlation for pccov_fut.

‘corr’: Take the correlation between emitot and emicov, unless it is too ‘bad’, then take the mean.

Default: ‘corr’.

all_sectors_covered(database, meta, ssp_pccov, info_per_country, info_for_iso, iso3, ssp, cov_gases, ssp_kyoto_ipcm0el)

If all sectors are covered, one can calculate the pccov_fut from the given share per gas and the gases that are covered.

not_all_sectors_covered(coverage, pccov_his, preference_pccov_fut, info_for_iso, iso3, ssp_kyoto_ipcm0el, ssp_pccov, ssp, info_per_country, available_years, first_year_for_slope, slope_lim, rvalue_lim)

If not all sectors are covered check for the slope of the regression line to pccov_his, and if pccov_his does not change too much use the mean (if preference_pccov_fut = 'mean'), else use the correlation between emitot and emicov.

data_to_database(ssp_pccov, database, meta, ssp, primap)

Put the data to 'database'. pccov, pcnotcov, emicov, and eminotcov.

> If all sectors are covered, one can calculate the pccov_fut from the given share per gas and the gases that are covered.
>
> If not all sectors are covered check for the slope of the regression line to pccov_his, and if pccov_his does not change too much use the mean (if preference_pccov_fut = 'mean'), else use the correlation between emitot and emicov.

Put the data to 'database'. pccov, pcnotcov, emicov, and eminotcov.

CHAPTER

ELEVEN

MODIFY_INPUT_HERE.INPUT_DEFAULT

Provide input for main_ndc_quantifications.py

UNITS: Units for time series: emissions in Mt CO2eq, population in Pers, GDP in 2011GKD. Units of NDC input data (infos_from_ndcs_default.xlsx): Mt CO2eq or tCO2eq / cap (all AEI targets are in emissions per capita).

PREPROCESSING: If you want to do the preprocessing of data, run preprocessing.py, and update the 'folder_preprocess' in setup_metadata.py. Else, the folder stored in setup_metadata.py will be used.

Chose the method for the pathway calculations (per country pathways).

'constant_percentages': The percentage difference to the baseline emissions of the last available target year is kept constant.

'constant_emissions': The emissions of the last available target year are kept constant.

Default: meta.method_pathways = 'constant_percentages'

For which countries should targets be used for the calculation of emission pathways for group of countries? For the others, the baseline emissions will be used.

countries: 'all', or e.g., ['EU28', 'AUS', 'CHN'], or e.g., get_isos_groups(['ANNEXI']).

Default: meta.calculate_targets_for = {'use_it': True, 'countries': 'all'}

Example: cat_countries = ['ARG', 'AUS', 'BTN', 'BRA', 'CAN', 'CHL', 'CHN', 'CRI', 'EU28',

> 'ETH', 'GMB', 'IND', 'IDN', 'JPN', 'KAZ', 'KEN', 'MEX', 'MAR', 'NPL', 'NZL','NOR', 'PER', 'PHL', 'RUS', 'SAU', 'SGP', 'ZAF', 'KOR', 'CHE', 'TUR', 'ARE', 'USA', 'UKR', 'VNM']

meta.calculate_targets_for = {'use_it': True, 'countries': cat_countries}

Which target-types should be prioritised in the calculation of group-pathways?

ndcs_type_prioritisations can be a certain target tpye (e.g., 'ABS'), or 'TYPE_ORIG' or 'TYPE_CALC'. Or several ordered options (only makes sense for != TYPE_ORIG and != TYPE_CALC).

One can chose from ['TYPE_ORIG', 'TYPE_CALC', 'ABS', 'RBY', 'RBU', 'ABU', 'REI', 'AEI'].

If TYPE_ORIG: use the 'original target type' (what has been stated (+/-) in the NDC as target type). If TYPE_CALC: use the target type that has been assessed to be the 'best suitable' (based on the NDC). Explanation: e.g., when it is an RBU target, but the absolute target emissions are available (e.g., given value, or based on their BAU and %-reduction), TYPE_ORIG can be RBU, and TYPE_CALC can be ABS. It can also have TYPE_ORIG is NGT and TYPE_CALC is ABU, as they quantified some reductions. Iterating through ndc_type_prioritisations, and using TYPE_CALC if none of the iterations found target values for the current target type in the NDC input file. If 'countries' is 'all', apply it to all countries. Else, give ISO3s, and it is only applied to those countries. Else, the pathway is calculated based on TYPE_CALC.

Default: meta.ndcs_type_prioritisations = {'use_it': True, 'ndcs_type_prioritisations': ['TYPE_CALC'], 'countries': 'all'}

Use NDC emissions data if available. If TYPE_CALC is used set it to True, for TYPE_ORIG set it to False.

For countries without unconditional target: use the baseline emissions for the unconditional pathway even if the conditional target is worse than the baseline (in 2030)?

Default: meta.use_baseline_for_uncondi_even_if_baseline_is_better_than_condi = False

The targets are strengthened by ndc_strengthen.

Chose between ‘how_to’: ‘add’ or ‘multiply’. ‘add’: the reduction is increased by adding the value in ‘pc’. ‘multiply’: the reduction is increased by multiplying with (100% + the value in ‘pc’).

If this results in a % reduction that exceeds 100%, it is set to 100% (meaning a total reduction of the covered part of emissions).

For absolute targets (ABS, ABU, AEI), it is not distinguished between ‘add’ and ‘multiply’.

Default: meta.strengthen_targets = {‘use_it’: False}

Predefine that the coverage used for the pathways is 100% for certain countries. Only possible for relative targets / reductions.

Default: meta.set_pccov_to_100 = {‘use_it’: False}

Groups for which to get the pathways. ‘EU28’, ‘EARTH’, ‘R5ASIA’, ‘R5LAM’, ‘R5MAF’, ‘R5OECD’, ‘R5REF’ will be calculated per default. The R5 regions are needed for the temperature pathways (PRIMAP Emissions Module / Climate Module).

One can chose some of the following groups: ‘ANNEXI’, ‘ANNEXI_KAZ’, ‘AOSIS’, ‘AR5’, ‘AG’, ‘BRICS’, ‘EIG’, ‘G7’, ‘G20’, ‘G77’, ‘GRADUATED_LDCS’, ‘IMO’, ‘LDC’, ‘LLDC’, ‘NON_ANNEXI’, ‘OECD’, ‘OPEC’, ‘SIDS’, ‘UMBRELLA’, ‘UNFCCC’, ‘UN_REGIONAL_GROUPS’, ‘AILAC’, ‘ALBA’, ‘APG’, ‘BASIC’, ‘CACAM’, ‘CD’, ‘CfRN’, ‘CVF’, ‘EEG’, ‘EIT’, ‘G77+China’, ‘GRULAC’, ‘KYOTO’, ‘LAS’, ‘LMDC’, ‘PA’, ‘SICA’, ‘UN’, ‘WEOG’ And provide a list for meta.groups_for_which_to_calculate_pathways

Default: meta.groups_for_which_to_calculate_pathways = []

CHAPTER TWELVE

_TO_BE_RUN

Script to run the NDC quantifications (main_ndc_quantifications). This includes per-country target emissions, pathways, and globally aggregated pathways.

Put in the name(s) of the wanted input-file(s) here. You can run several input-files in a row.

Default: main_ndc_quantifications(‘input_SSP2_typeCalc_pccov100’, ‘’)

CHAPTER

THIRTEEN

MAIN_NDC_QUANTIFICATIONS

main_ndc_quantifications(input_file, lulucf_prio)

Calculation of NDC mitigation target emissions. Main file.

————————-

Input examples:

input_file = ‘input_SSP2_typeCalcForAllCountries’ (name of input-file, stored in /MODIFY_INPUT_HERE).

lulucf_prio = ‘’ or ‘UNFCCC’ or ‘FAO’.

- If it is ‘’ the default LULUCF prioritisation is used (CRF, BUR, UNFCCC, FAO).
- For ‘UNFCCC’: UNFCCC, CRF, BUR, FAO.
- For ‘FAO’: FAO, CRF, BUR, UNFCCC.

————————-

Per country, target year, target type, conditionality, and range, the target emissions and emissions pathways are calculated. The target is calculated once including, and excluding LULUCF.

————————-

Target types

ABS: Absolute target emissions. E.g., target is to reduce emissions in 2030 to 500 MtCO2eq.

RBY: Relative reduction compared to base year. E.g., 20% emissions reduction compared to 2010 emissions in 2030.

RBU: Relative reduction compared to BAU. E.g., 20% emissions reduction compared to business-as-usual (BAU) emissions in 2030.

ABU: Absolute reduction compared to BAU. E.g., 350 MtCO2eq reduction compared to BAU emissions in 2030.

REI: Relative emissions intensity reduction. Compared to base year. E.g., 20% emissions intensity reduction compared to 2010 emissions intensity in 2030. Or compared to BAU. This is basically a simple RRB target, but some NDCs state it as intensity targets.

AEI: Absolute emissions intensity. E.g., 2.1 tCO2eq/cap in 2030.

NGT: Non-GHG targets. Nothing is calculated, baseline emissions are assumed.

————————-

Per country one target type is chosen for the aggregation to a global pathway. This is generally the type_orig or type_calc.

type_orig: what is said (+/-) in the NDC, e.g., 20% reduction compared to BAU (RBU). In this case the comparison emissions are prioritised (comparison runs with 'external' input data).

type_calc: what seems more suitable for the pathway calculations, e.g., if for the BAU target a quantification is given in th In this case the emissions given in the NDCs are prioritised.

——————————-

Globally and regionally aggregated emissions pathways are given with the output suitable for the MATLAB PRIMAP Emissions / Climate Module. They can be used to derive 2100 temperature estimates corresponding to the calculated 1990-2030 pathways.

Read input time series 1990-2050. For all countries in meta.isos.EARTH. For the EU28 countries, the targets are calculated separately (using the EU28 NDC info for each of the countries). In the end, a pathway is calculated for EU28.

Read tables: '*KYOTOGHG_IPCM0EL_TOTAL_NET_*' + meta.ssps.chosen + 'FILLED_PMSSPBIE' 'KYOTOGHG_IPCMLULUCF_TOTAL_NET_INTEREXTRAPOL_VARIOUS{lulucf_prio}' '**KYOTOGHG_IPCM0EL_COV_PC_**' + meta.ssps.chosen + 'FILLED_COVERAGE' 'KYOTOGHG_IPCMLULUCF_COV_EMI_HISFUT_COVERAGE' '**POP_DEMOGR_TOTAL_NET_**' + meta.ssps.chosen + 'FILLED_PMSSPBIEMISC' '**GDPPPP_ECO_TOTAL_NET_**' + meta.ssps.chosen + 'FILLED_PMSSPBIEMISC'

In principle one can also provide other tables (same type of data, but from other sources / with other values), as long as they have the same structure and names.

If it is type_calc: use the ndc-emissions if available. For type_orig: use the comparison data.

Additionally calculate and add the following tables to the database: '**KYOTOGHG_IPCM0EL_NOTCOV_PC_**' + meta.ssps.chosen + 'FILLED_COVERAGE' '**KYOTOGHG_IPCM0EL_COV_EMI_**' + meta.ssps.chosen + 'FILLED_COVERAGE' '**KYOTOGHG_IPCM0EL_NOTCOV_EMI_**' + meta.ssps.chosen + 'FILLED_COVERAGE' '**KYOTOGHG_IPC0_TOTAL_NET_**' + meta.ssps.chosen + '_VARIOUS{lulucf_prio}'

Write relevant input-info to log_file.md in the output-folder. In this file one will find the necessary information to re-run the calculations with the same setup.

Correct or modify the calculation options given in input_file if necessary. ndcs_check_options_for_target_calculations(meta).

TARGETS Calculate the target emissions per country and un/conditional & best/worst & year. ndcs_calculate_targets(database, meta).

PATHWAYS Country-pathways.

Calculate the emissions pathways for un/conditional_best/worst targets, per country. For countries without targets use the baseline emissions (if available). ndcs_calculate_pathways_per_country(database, calculated_targets, meta)

PATHWAYS Group-pathways.

Calculate the emissions pathways for un/conditional_best/worst targets, per group of countries. ndcs_calculate_pathways_per_group(pathways_per_country, meta)

CHAPTER

FOURTEEN

MAIN_FUNCTIONS.NDCS_SOME_INITIAL_TESTING

ndcs_some_initial_testing(database, meta)

Check input data for ‘how many countries have no / missing data’ and ‘are the values of pc_cov/ncov between 0 and 1’. Does not check whether the emissions / pop / gdp data seem realistic.

CHAPTER

FIFTEEN

MAIN_FUNCTIONS.NDCS_CHECK_OPTIONS_FOR_TARGET_CALCULATIONS

ndcs_check_options_for_target_calculations(meta)

Check whether the values for attributes of classes meta.calculate_targets_for, meta.ndcs_type_prioritisations, meta.set_pccov_to_100, and meta.strengthen_targets – given in input_file – are ok.

check_countries(countries, meta)

Check for valid country codes.

CHAPTER

SIXTEEN

MAIN_FUNCTIONS.NDCS_CALCULATE_TARGETS

ndcs_calculate_targets(database, meta)

> Calculate the NDC targets for each of the target types available for a country. Some NDCs give more than one target type (e.g., they give the absolute value, but say that it is a RBU target and therefore give the % reduction against BAU).
>
> EU28 countries are calculated separately, using the NDC information from the EU28 NDC. The per-country target emissions for EU28 countries does not equal the 'real' emissions targets (each of the countries has its own targets to in-sum get to the EU28 total target).

check_ndc_values(ict, iso_act)

> Check if ndc_values seem ok. For the different target types there are some criteria to check if the ndc_value seems plausible. Does not cover every eventuality.

strengthen_targets(meta, ict, ndc_value_exclLU, ndc_value_inclLU, iso_act)

> Apply a strengthening to the targets. Depends on 'how_to' ('multiply' or 'add').
>
> 'add': the reduction is increased by adding the value in 'pc'. 'multiply': the reduction is increased by multiplying with (100% + the value in 'pc'). If this results in a % reduction that exceeds 100%, it is set to 100% (meaning a total reduction . . .).
>
> For absolute targets (ABS, ABU, AEI), it is not distinguished between 'add' and 'multiply'.

quantification_per_country(iso_act, meta, lulucf_first_try, calculated_targets, txt)

> Per country get the information on the different available target types, and get the calculation data for emi, pop, gdp for the reference and target years. Calculate the single target emissions, stored in ndc_targets.csv.
>
> > Get all available target years & un/conditional & best/worst targets. ndc_value depends on the target type, either it is an absolute emission (ABS, ABU), a percentage for relative reductions (RBY, RBU, REI), or an emissions intensity (AEI).
> >
> > - ABS: target emissions in MtCO2eq.
> > - ABU: absolute reduction in MtCO2eq (e.g., -3500 stands for 3500 MtCO2eq reduction).
> > - RBY, RBU, REI: percentage reduction (e.g., -20 stands for 20% reduction).
> > - AEI: emissions intensity in target year, e.g., 3.2 stands for 3.2 t/cap or 3.2 t/GDP, when int_ref is POP or GDP, respectively.
> >
> > In infos_from_ndcs_default.xlsx, infos_from_ndcs.csv, and meta.ndcs_info, the 'available targets' per target type are stored as nested dictionaries and can be read in using json. Targets are also separated in 'inclLU' and 'exclLU'.
> >
> > > Iterate through the combinations of target years & un/conditional & best/worst. Do the calculations depending on the ndc values and the target type.

Set the coverage to 100% (not changing LULUCF), if meta.set_pccov_to_100 is True and the country is in the 'wanted' countries. Only for relative targets / reductions.

Check if ndc_values seem ok. For the different target types there are some criteria to check if the ndc_value seems plausible. Does not cover every eventuality.

Get the numerical values from ndc_value_exclLU and ndc_value_inclLU. exclLU or inclLU was assessed based on the NDCs and stored together with the targets (reductions or absolute values). exclLU was chosen if LULUCF is not covered, or if LULUCF was covered, but it was stated that this value is for exclLU. inclLU was chosen if LULUCF is covered, or if LULUCF was not covered and it was stated that this value is for inclLU.

Convert the given absolute value to AR4 (no conversion factors for AR5 available). Based on national conversion factors from AR2 to AR4 from PRIMAPHIST21 KYOTOGHG_IPCM0EL.

If strengthen NDC is chosen apply the strengthening to the reductions.

Calculate the mitigated target year emissions depending on the target type.

calculate_targets_depending_on_type(ict, iso_act, refyr, taryr, ndc_value_exclLU, ndc_value_inclLU, meta, lulucf_first_try, txt)

Calculation of targets: excluding LULUCF and including LULUCF. For one country, target type, target year, conditionality, range.

Get the emissions values from NDCs.

Calculate the NDC-level for relative targets. E.g., target is -20%, so the level is 80% (100%-20%).

Get the information on whether LULUCF is included in the target or not.

Get the LULUCF data for the reference and target year.

calc_targets_inclLU_exclLU(tar_emi_exclLU, tar_emi_inclLU, bl_onlyLU_taryr, tar_type_used, txt)

If not both, inclLU and exclLU information are given: calculate the 'other case' from the given case.

If no inclLU target is given, calculate it as the sum over tar_emi_exclLU and bl_LU.

Options to calculate the exclLU target from the given inclLU target:

meta.prio_tar_exclLU_from_tar_inclLU == 'apply_rel_red_to_emi_exclLU':
If no exclLU target is given, calculate it assuming the same % reductions in all sectors. Calculate the % reduction of ABS_inclLU compared to the bl_inclLU (use the value given in NDC, if possible) and apply it to our emi_bl_exclLU (assuming 100% coverage).

meta.prio_tar_exclLU_from_tar_inclLU == 'subtract_LU_in_taryr':
If no exclLU target is given, subtract the target year onlyLU estimate from the inclLU target (like CAT).

If it needs a second try due to the LULUCF part (if the exclLU target is negative, when calculate from the inclLU target):

Get the ABU_inclLU and split it into the onlyLU and exclLU parts (depending on the onlyLU and exclLU contributions in the target year).

tar_exclLU: bl_exclLU + ABU_exclLU (ABU_exclLU is negative).

> ABS target: tar_emi is the given absolute emissions value.
>
> ABU target: tar_emi is the given baseline minus the absolute reduction.
>
> AEI target: tar_emi is the given absolute emissions intensity multiplied by the target year GDP or POP.
>
> RBY, RBU or REI target.
>
> > REI compared to BAU is same as RBU. For REI_RBY, the emissions intensities must be used.
> >
> > inclLU, LU is covered: include the emi_bl_onlyLU_refyr emissions in the target. If emi_bl_onlyLU_refyr is negative, only apply the reduction to the exclLU-part. And add the bl_onlyLU_refyr as is.

Iterate through the iso3s in meta.ndcs_info (EU28 as single countries, using the EU28 target info). Calculate ‘all available targets’ per country (targets as in meta.ndcs_info, in columns [‘ABS’, ‘RBY’, ‘RBU’, ‘ABU’, ‘REI’, ‘AEI’]).

> First run of quantifications.
>
> If any of the tar_exclLU are negative re-run the quantifications with different method.

CHAPTER

SEVENTEEN

MAIN_FUNCTIONS.NDCS_CALCULATE_PATHWAYS_PER_COUNTRY

ndcs_calculate_pathways_per_country(database, calculated_targets, meta)

Calculate the pathway per country and target (per un/conditional best/worst). Only for IPCM0EL.

Stored in 'ndc_targets_pathways_per_country.csv'.

add_timeseries(meta, database, ptws_all)

Get the cydata for all countries (are stored in ts_act.data) and add some information (gwp, category, . . .).

add_targets_from_other_conditionality_if_needed(meta, targets_act)

If there is a value in unconditional_best in a year, but not in unconditional_worst, put it there. If there is a value in conditional_best in a year, but not in conditional_worst, put it there. If there is a value in unconditional in a year, but not in conditional, put it there (the one stored in 'unconditional_worst').

calculate_pathways(ptws_all, emi_bl_act, targets_act, current_tar, curr_tar, iso_act, meta)

Calculate the pathways per un/conditional_best/worst, assuming a linear in/decrease of the % difference to the baseline emissions. E.g., if the country has one unconditional_best target that equals a reduction of 20% in 2030 (compared to baseline emissions): the pathway has a 0% reduction in meta.years.pathways[0], and in 2030 the given 20% reduction, with a linear increase between meta.years.pathways[0] and 2030.

Use the baseline emissions for 2020. Calculate the percentage level (how much percent of the baseline emissions do the single target emissions represent). pc_level: interpolate between available values, and keep the last available value of pc_level constant (if meta.method_pathways == 'constant_percentages'), or keep the last emissions level constant (if meta.method_pathways == 'constant_emissions').

E.g., 2020 is 100%, 2030 is 80%. For 2025 it is 100% + (80%-100%)/(2030-2020) * (2025-2020) = 90%. All values of pc_level are 100% (= baseline emissions) if there are no available_years. baseline emissions are used in years before the first meta.years.pathways. Apply pc_level to baseline emissions to calculate the pathway.

> Keep pc_level constant after last available target for targets that are below the baseline of that year.

If the last target is higher than the baseline, use the same growth as in the baseline. Else, the mitigated pathway can increase a lot.

Keep the emissions constant after last available target if meta.method_pathways == 'constant_emissions'. Else the pc_level is kept constant.

If the country has an entry for 'DECLINE_AFTER_YEAR': stop the mitigated emissions from increasing after that year.

add_baseline_emissions(emi_bl_act, meta, iso_act, columns)

Add the baseline emissions for each iso3 (as un/conditional best/worst pathways).

Get the cydata for all countries (are stored in ts_act.data) and add some information (gwp, category, . . .).

Iterate through EARTH countries, EU28 as single countries. Calculate the pathway per country and target type and un/conditional best/worst. Use baseline emissions where necessary. Use the unconditional target as conditional target as well, if the country does not have a conditional target.

If meta.use_baseline_for_uncondi_even_if_baseline_is_better_than_condi = True use the baseline emissions for the unconditional pathways even if the conditional pathway (for 2030) is worse than the baseline. Else use the conditional pathway as unconditional pathway as well.

If nothing is available, use 'baseline_emissions'.

Calculate pathways per target type.

If there is a value in unconditional_best in a year, but not in unconditional_worst, put it there. If there is a value in conditional_best in a year, but not in conditional_worst, put it there. If there is a value in unconditional in a year, but not in conditional, put it there (the one stored in 'unconditional_worst').

Calculate the pathways per un/conditional_best/worst, assuming a linear in/decrease of the % difference to the baseline emissions. E.g., if the country has one unconditional_best target that equals a reduction of 20% in 2030 (compared to baseline emissions): the pathway has a 0% reduction in meta.years.pathways[0], and in 2030 the given 20% reduction, with a linear increase between meta.years.pathways[0] and 2030.

Check for meta.use_baseline_for_uncondi_even_if_baseline_is_better_than_condi. If it is False and the conditional_worst pathway is worse in 2030 than the baseline put the conditional_worst pathway to the unconditional pathways.

For each country, even if it does have a different target, also give out 'baseline_emissions'.

CHAPTER EIGHTEEN

MAIN_FUNCTIONS.NDCS_CALCULATE_PATHWAYS_PER_GROUP

ndcs_calculate_pathways_per_group(pathways_per_country, meta)

Calculate the emissions pathways for a group of countries by summing up all available per-country pathways, per un/conditional_best/worst (chosing one of the target types per country). And using baseline emissions where no target is available.

Chosen per-country pathways stored in ndc_targets_pathways_per_country_used_for_group_pathways.csv. Group pathways stored in ndc_targets_pathways_per_group.csv.

get_pathways_per_country_used_for_group_pathways(input_pathways, pathways_per_country, meta)

Get the 'chosen' country-pathways that will be used for the group pathways. Depend on the preferred (and available) target types.

Chosen per-country pathways stored in ndc_targets_pathways_per_country_used_for_group_pathways.csv.

calc_pathways_of_group(input_pathways, ptws_groups, ptws_chosen_per_country, meta, group_act, tars_to_use)

Calculate the group pathways (for un/conditional best/worst; using the chosen country-pathways).

CHAPTER

NINETEEN

MAIN_FUNCTIONS.NDCS_SOME_FINAL_TESTING

ndcs_some_final_testing(database, meta)

Doing some testing of the constructed global pathways.

check_one()

Check if the EARTH pathways for emi_bau, pop, gdp seem ok.

check_two()

Check if the EARTH pathways for pc_cov, emi_cov seem ok.

check_three()

Check if the ‘used’ EARTH pathways seem ok.

check_four()

Check if the target pathways for EARTH seem ok.

CHAPTER

TWENTY

NDCS__TARGET_CALCULATIONS_FOR_INPUT_XLSX

Countries for which something (some emissions based on data provided within the NDC or target emissions) was calculated in ndcs__target_calculations_for_input_xlsx: AFG, AGO, ALB, AND, ARG, ARM, AZE, BDI, BEN, BGD, BIH, BLR, BRB, BTN, BWA, CAF, CHE, CHL, CIV, CMR, COD, COG, COK, COL, COM, CRI, CUB, DJI, DMA, DOM, ECU, ERI, FJI, FSM, GAB, GEO, GHA, GIN, GMB, GTM, HND, HTI, IDN, IND, ISR, JAM, JOR, JPN, KEN, KHM, KIR, KOR, LAO, LBN, LCA, LIE, LKA, LSO, MAR, MCO, MDA, MDG, MDV, MEX, MHL, MKD, MLI, MNG, MOZ, MRT, MUS, MWI, MYS, NAM, NER, NGA, NOR, OMN, PAK, PER, PRY, PSE, RWA, SEN, SLB, SMR, STP, SYC, TCD, TGO, THA, TJK, TLS, TUN, TZA, UGA, UKR, URY, USA, VCT, VNM, WSM, ZMB.