https://doi.org/10.5194/gmd-2020-138
Preprint. Discussion started: 28 May 2020

Silicone v1.0.0: an open-source Python package for inferring missing emissions data for climate change research

Robin D. Lamboll[1], Zebedee R. J. Nicholls[2,3], Jarmo S. Kikstra[4], Malte Meinshausen[2, 3] Joeri Rogelj[1]

[1]Grantham Institute for Climate Change and the Environment, Imperial College London, UK
[2] Australian-German Climate and Energy College, The University of Melbourne, Parkville, Victoria, Australia
[3] School of Earth Sciences, The University of Melbourne, Parkville, Victoria, Australia
[4] International Institute for Applied Systems Analysis, Laxenburg, Austria

Correspondence to: Robin Lamboll (r.lamboll@imperial.ac.uk)

Abstract. Integrated assessment models (IAMs) project future anthropogenic emissions for input into climate models. However, the full list of climate-relevant emissions is lengthy and most IAMs do not model all of them. Here we present silicone, an open-source Python package which infers anthropogenic emissions of missing species based on other known emissions. For example, it can infer nitrous oxide emissions in one scenario based on carbon dioxide emissions from that scenario plus the relationship between nitrous oxide and carbon dioxide emissions in other scenarios. This broadens the range of IAMs available for exploring projections of future climate change. Silicone forms part of the open-source pipeline for assessments of the climate implications of IAMs by the IAM consortium (IAMC). A variety of infilling options are outlined and their suitability for different cases are discussed. The code and notebooks explaining details of the package and how to use it are available from the GitHub repository, https://github.com/GranthamImperial/silicone. There is an additional repository showing uses of the code to complement existing research at https://github.com/GranthamImperial/silicone_examples.

1. Introduction

1.1. General context and problem setting

Integrated assessment models (IAMs) are scientific modelling tools that integrate knowledge from different academic disciplines with the aim to explore and inform policy decisions (IPCC, 2014b, 2018; Weyant, 2017). They are widely used in climate change research to combine insights from energy, economy, agricultural, and natural sciences, with the aim to create scenarios that explore how societal decisions can affect projected greenhouse gases and other emissions, as well as their related climate outcomes (Huppmann, Rogelj, Kriegler, Krey, & Riahi, 2018; IPCC, 2014a; Riahi et al., 2017; Rogelj et al., 2018). However, IAMs do not always exhaustively represent all possible processes or sources of climate-relevant emissions. Thus, many IAM scenarios lack projections for some climate forcers, be it specific greenhouse gas emissions or aerosol precursors. A complete set of these climate forcers is required to accurately estimate the overall climatic effects of a given scenario (Meinshausen, Raper, & Wigley, 2011; Smith et al., 2018), as a large number of supposedly minor emissions may exert a significant radiative forcing between them (IPCC, 2013; Meinshausen et al., 2017; O'Neill et al., 2016). Scenarios that only report a limited set of greenhouse gases or climate forcers thus must be complemented by estimated evolutions of missing emissions derived without further economic analysis. We term this estimation 'infilling'. If no infilling is attempted, the unevaluated emissions are effectively infilled with zeros, which will clearly create systematic biases and potential artefacts in the projected temperatures. Depending on the radiative forcing of the species in question, this bias may be positive or negative, so infilling with zeros would not necessarily be a conservative choice. Most earlier studies overcame this problem in one of two ways: with expert-based ad-hoc decisions on how to adequately fill-in missing species (Schaeffer et al., 2015); or by assuming that a pathway will occur at the same quantile for each set of emissions in a particular year, although the quantile can vary over time (Meinshausen et al., 2006). However, the former clearly does not scale easily, and the latter approach, termed the "Equal quantile walk" (EQW), ignores trade-offs and specific relationships between emission species resulting from how competing technologies, behaviours and industrial practices result in different emissions. A more sophisticated "Generalized Quantile Walk" technique can capture the effect of trade-offs and was recently introduced in section 3.8.1 in (Meinshausen & Dooley, 2019), involving quantile regression between a lead variable (fossil CO2 emissions) and other gases for every individual year. Unfortunately, the implementation there did not consistently guarantee that higher quantiles resulted in higher emissions, and has not been followed up with any peer-reviewed work that does so.

Here we present a new toolbox of methods to address these recurring infilling challenges in the climatic assessment of socioeconomic emissions scenarios. The toolbox introduces new approaches as well as building on and combining previous approaches. It significantly improves the codebase in terms of flexibility, applicability, reproducibility and versatility.

1.2. The aim of Silicone

Silicone is a Python package designed to enable users to expand scenario projections of a limited set of climate forcers to a full set required for an accurate climatic assessment. In essence, its methods are grounded in a comparison of the co-evolution of anthropogenic emissions in scenarios that are readily available in the literature (Huppmann et al., 2018; Riahi et al., 2017; Rogelj et al., 2018). Silicone aims to provide IAM teams that do not represent all individual climate forcers with robust methods to complement their model output and facilitate a climatic assessment of their work. Furthermore, Silicone also aims to provide geoscience researchers with a tool to easily develop stylized, yet internally consistent future emission pathways of the most important climate forcers. It can also estimate or calculate missing sectoral data. Notebooks describing how to use these tools are available on the accompanying GitHub repository (Lamboll, Nicholls, & Kikstra, 2020b) and the formal documentation is available at (Lamboll, Nicholls, & Kikstra, 2020a). Additional examples of using this for the specific situations outlined below are included in a separate GitHub repository (Lamboll, 2020). The package is open-source and intended to allow groups to write their own infilling methods if desired. Users and collaborators are invited to submit their code for review by the team on GitHub.

Silicone is compatible with a suite of tools that make up the IAM climate assessment pipeline developed under the umbrella of the Integrated Assessment Modelling Consortium (IAMC). This pipeline includes tools to manipulate and plot IAM data (pyam, (Gidden & Huppmann, 2019)), harmonise mismatches in historical emissions (aneris, (Gidden, Fujimori, et al., 2018)). The estimation of climatic impact is performed by OpenSCM, managed by the OpenSCM community (Nicholls, Gieseke, Lewis, & Willner, 2020), which is compatible with the data structure of the pipeline. This pipeline is being developed in support of the IAM community and the IAM scenario assessment for the forthcoming Sixth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC AR6) in particular.

2. Methods

Silicone takes a database that contains data for at least two emissions species (this database is referred to as the 'infiller' database) and derives a relationship between these timeseries. It then applies that relationship to a second database (the 'infillee' database), which does not have any data for one of the emissions species in the infiller database. For example, based on an infiller database of CO_2 and N_2O emissions, silicone could then derive N_2O emissions compatible with the CO_2 emissions in a less complete infillee data base.

Silicone offers a range of tools that apply methods for doing this infilling which are appropriate in different circumstances, depending on the amount of input data and how much we know about the narrative behind our emissions. These tools are referred to as 'crunchers'. Each of these crunchers takes a 'lead variable', found in both the infiller and infillee databases, and uses it to infer the value of a 'follower variable', found only in the infiller database (hence missing in the infillee database). There are also several tools for easily infilling multiple variables, called 'multiple infillers'. These may have multiple follower or lead variables.

2.1. Methods overview

2.1.1. Cruncher guide

There are seven types of cruncher. These are outlined in Table 1 below. A flow chart to guide the choice is shown in Figure 1. There is also a series of notebooks with examples of how to use them all in the main GitHub repository (Lamboll et al., 2020b).

2.1.2. Multiple Infiller and Aggregation Tools Guide

Multiple infillers are for cases where there are relationships between multiple values for inputs or outputs that need to be considered at the same time. They allow less tailored approaches to infilling but can ensure that the infilling is faster or more consistent than infilling each of the variables separately. These are outlined in Table 2.

2.2. Mathematical detail

Notebooks presenting benefits and risks of each cruncher type can be found in the silicone GitHub (Lamboll et al., 2020b) and may be useful to have as examples when analysing the work below, as well as demonstrating how to use them.
There are two main classes of infillers: those based on ratios between two emission pathways and those based on the absolute emission values in the infiller database.
The ratio-based approaches are better for cases where the lead values to be infilled are outside the range in the infiller database and we expect emissions to scale together. For instance, if we are infilling regional data from aggregate data, or splitting up aggregate emissions into their component gases. However, care needs to be taken when infilling emissions that are non-negative using a lead value that may be of any sign. In that case, the ratio method might produce values for the infillee emissions that are unsupported by any available evidence. Singular behaviour may also be encountered when the average lead data is close to zero in the infiller database. The different crunchers present different ways to estimate the ratio to use.
The absolute value-based techniques infill with values derived from the absolute data found in the infiller database, or linear combinations of them. This means that they will always return values within the range spanned by the infiller database. This is most appropriate for processes where we have a larger number of cases, preferably with similar lead emissions in the infiller database or where we expect the follower emissions to be strongly bounded rather than increasing in line with other variables. They may be considered more stable and more conservative. The quantile rolling window (QRW) cruncher can be used in either ratio or absolute (non-ratio) mode, the absolute mode being the default.

2.2.1. Ratio infilling methods

These methods all firstly estimate the ratio of the lead gas to the follower gas. In all cases, we first estimate the ratios, written as $R(t)$ at time t. Once these have been calculated, the follower value in infillee database, $E_f(t)$, is valued as $E_f(t) = R(t)E_l(t)$, where $E_l(t)$ is the lead value in the infillee database. 'Constant ratio' and 'latest time ratio' methods both use the same ratio for all infill times, $R(t) = R$. The ratios may still be different for each different timeseries in the infillee database. With the 'constant ratio' method, the ratio must be given as an input parameter. The 'latest time ratio' method uses the ratio between the mean follower data in the infiller database (we denote this database with lower-case, e_f) and the value of lead variable in the infillee data (E_l), both evaluated at the latest time for which there exists follower data in the infiller database, t_{last}. This assumes that we have historic data up until sometime, after which it stops. This gives us the equation $R = \frac{\langle e_f(t_{last})\rangle}{E_l(t_{last})}$, where the angular brackets mean taking the (algebraic) mean with equal weighting for all estimates (typically historical estimates) and the lower case $e_f(t)$ represents the follower values in the database at time t. This ensures that at t_{last}, all infilled data will infill $E_f(t_{last}) = \langle e_f(t_{last})\rangle$.

The 'time-dependent ratio' method relies on having data in the infiller database for all times, and allows the ratio to vary with time. For this we use $R(t) = \frac{\langle e_f(t)\rangle}{\langle e_l(t)\rangle}$. Optionally, the averaging can be taken only over cases where the sign of the lead variable is the same in both the infiller and infillee case – this will guarantee that the infilled value takes the same sign as that of follower values in the database. A useful property of this relation is that if in every scenario averaged over, the emissions of several substances sum to another substance, e.g. if $e_1 = e_2 + e_3$, then $\langle e_1\rangle = \langle e_2\rangle + \langle e_3\rangle$ and so $1 = \frac{\langle e_2\rangle}{\langle e_1\rangle} + \frac{\langle e_3\rangle}{\langle e_1\rangle}$. This means when the aggregate is the lead and the components are follows, the sum of the two ratios is one, so we can use this infiller to break an aggregate value into components and know that the total is conserved. This relationship generalises to any number of components, still holds when emissions can be negative, and is irrespective of whether the averaging includes all values or only those where the lead has a particular sign. This is the foundation for the 'decompose collection' multiple infiller. It does however rely on all scenarios having values for all of these variables, so constructs a new version of the aggregate variable in case there is an inconsistency or absent data in the infiller database. This is not performed by the cruncher, so missing or inconsistent data can potentially distort results here.

The 'quantile rolling window' method may be applied in ratio mode, in which case we calculate $R(t)$ by calculating the ratio for each scenario, $r_s(t) = e_f(t)/e_l(t)$. This method finds quantiles of the ratio in the infiller database at set points along the range of lead values in the infiller database. It is more fully described below in the absolute value section.

2.2.2. Absolute value infilling methods

The 'RMS closest' cruncher filters the infiller database for models with all the same times as the infillee data. It then ranks models and scenarios by the root mean squared (RMS) difference between the lead data in the infiller and infillee database, with the average being taken over all times. It returns the follower data from the scenario/model combination with the smallest RMS difference. In the case of a draw, the value that occurs earlier in the infiller database will be used. It is the only infiller that is not time-independent, i.e. the value of the lead at one time impacts the whole timeline.

The 'linear interpolation' method constructs an (unsmoothed) linear interpolator function between all lead and follower points in the infiller database at a given point in time. If multiple points have exactly the same lead value, the average follow value is used. The follower value returned is then the interpolated value for the infillee lead. The 'Interpolate specified scenarios and models' cruncher filters for scenarios and models that match a given text string before performing the same action of the linear interpolation cruncher.

The 'quantile rolling windows' cruncher, applied with the default option 'use_ratio=False', infills the values based on interpolating between quantiles calculated at fixed points across the range of lead values in the infiller database. This is identical to the above discussion where use_ratio=True, except using the actual follower values instead of the ratios between lead and follow. It is inspired by the Generalized Quantile Walk approach in section 3.8.1 of (Meinshausen & Dooley, 2019). An illustration of the idea behind this cruncher is shown in Figure 2: Schematic of how the quantile rolling window cruncher determines the follow value to use. a) Example relationships between lead (CO_2) and follow (CH_4) variables over time. b) Five windows are drawn and a weighting function constructed for each window. c) A relationship between the sum of the weights and the follow value is established and the follow value at the desired quantile is returned. . For each time in the infiller database, it splits the range of lead values into $n_{windows}$ points with values e_p, including the highest and lowest values. For each window, the weightings of each point are given as $w_p(e_l) = 1/(1 + \left(\left(e_p - e_l\right)/_{d_l}\right)^2)$, where d_l is the decay length, which defaults to half the separation between e_p. Amongst other things, this is a clear improvement over the Generalized Quantile Walk approach, as the latter uses equal weights within a fixed window of a certain fraction of the infiller database's lead values in a certain year. These values are then normalised so that $\sum w_p = 1$ and sorted into ascending order by e_f. The follow value at quantile q, evaluated at lead point e_p, is where the quantile equals the sum of weights of all smaller e_f plus half the weight of e_f itself, $q = \sum_{r_i<r_p} w_p(r_i) + \frac{w_p(r_p)}{2}$. Quantiles between these known points are evaluated by linearly interpolating this relationship. To infill a point at E_l, we interpolate between the known points e_p. Quantile crossing is not possible in this framework because at any given evaluation point higher quantiles cannot have lower values, and only linear fits between these points are used.

3. Rank Correlations

The infilling method is important. However, equally important is the choice of lead variable. The best choice is where there is a causal link between the lead and follower variable, particularly if there is a clear understanding of the implications of this link for the relative behaviour of the two variables. In most cases, there is no such certainty, and the best choice is then to find the lead variable with the best predictive power. We estimate this by the Spearman's rank correlation coefficient, a measurement of the monotonicity of the relationship between the two variables. In cases where this value is low, we anticipate the need for higher effort to select relevant cases from the infilling database. We use the data from the IPCC Special Report on Global Warming of 1.5 °C (Huppmann et al., 2018) as our database of scenarios and compare the

https://doi.org/10.5194/gmd-2020-138
Preprint. Discussion started: 28 May 2020

correlations between the different variables. The Silicone package has a function in the statistics section called 'calc_all_emissions_correlations', which will produce tables of both the Spearman (rank) and Pearson correlation coefficients, calculated separately for each year requested and also the time-averaged magnitude of the correlations. Since there is no reason to expect the relationships between variables to be linear, we will focus on the rank correlation in this analysis. All the crunchers work just as well with negative trends as with positive, so the sign of the correlations is not relevant for considering goodness of fit. Using this tool, we can calculate the decadal-averaged magnitude of the rank correlation coefficient, found in Table 3. We also calculate the variation of this value with time, and in cases where this exceeds 0.03, colour the cells orange. This is to indicate cases where more care needs to be taken to ensure that values are representative for the times of interest.

The immediate observation from the study of absolute rank correlations is that there is no clear, overall best infiller gas. CH_4 has a slightly higher average than other emissions and is reported by most models. CO_2 is reported by all models and has the second highest correlation, however this is somewhat inflated by having two of its constituents listed separately (a similar concern can be raised about F-gases). Generally these two therefore are the best choices for a 'default lead variable'. However there are some specific cases where the correlations are low, and much better choices could be made.

There is a cluster of values BC, CO, and OC, that correlate poorly with other emission pathways. Physically, these relate to incomplete burning, and are best infilled using each other. The F-gases, SF_6, HFCs, and PFCs also primarily relate to each other. Fortunately many models report F-gas emissions, so infilling these should best be done by splitting the F-gas basket into its constituents. Otherwise the default infillers, CO_2 and methane, should do reasonably.

4. Results

4.1. Reconstructing data

The choice of cruncher to use in different situations will depend on the expectations about the specific emissions in question. However, in cases where there are no clear expectations, it is good to have a default. In this section we assess to which degree the cruncher reproduces the follower data from one model and scenario given the lead data from that case and all data from all the other model/scenario combinations in the SR1.5 database. We try this with both methane and CO_2 as our lead variables and find similar results. Since we have no reason to believe that the answers will always scale with these variables (and ratio-based infillers have potentially unbounded errors), we choose four absolute-value infillers as our options: QRW (using only default settings, so in absolute mode), RMS closest, EQW and linear interpolation. We perform the infilling for each model/scenario combination, for each decade from 2020 to 2100, and report the root mean squared difference between the original value and the infilled value, normalised by the total variation in the follower value in the database at that time.

We see that in both cases with this fairly large infiller database, the approach that generates non-CO2 pathways most similar to those removed from the initial scenarios is the RMS technique, and that the QRW technique is fairly similar. Linear interpolation is expected to produce a noisy fit when given a large input dataset, so its performance is unsurprisingly worse. The Equal Quantile Walk (EQW) performs similarly poorly, due to effectively ignoring the relationship between the lead and follower data. A Shapiro-Wilks test indicates that neither CO_2 nor methane are statistically significantly non-Gaussian in distribution, either when analysed separately for each cruncher or as an aggregate. The small differences in rank between methane and CO_2 manifest in slightly lower values for methane. Performing a student's t-test on the results indicates that this result is statistically significant for the data as a whole (relative t-test t-statistic 3.4, p = 0.0013), although when considering each of the crunchers individually, only the RMS closest cruncher is significantly better with methane than CO_2 (p-values for QRW = 0.33, RMS Closest = 0.0098, linear interpolation = 0.078, EQW = 0.37). We therefore conclude that using either as the default will produce reasonable results, with this slight preference for using methane as our default infiller counteracted by the slightly lower availability of data (412 cases rather than 414, as seen in table 3).

We perform similar pairwise t-tests on the results of different crunchers, and find that the ordering of mean errors, (RMS closest < QRW < Linear interpolation ≈ EQW) are all statistically robust. The p-values are < 0.001 for all pairs except linear interpolation and EQW, which are much greater than 0.1, whether the comparison uses CO_2 lead data, or methane lead data, or both data combined for all cases. We stress that this does not always mean that the RMS closest technique is the best default, as it makes the assumption that the pathway being infilled is similar to a whole pathway found in the database. The advantage of the quantile rolling windows technique is its conservativity – that it tends to produce values more towards the median value – and time-independence, whereas RMS closest is better at reconstructing the data and has better consistency over time. Linear interpolation and EQW are best used in cases where there is a large degree of knowledge about the expected relationship between variables.

4.2. Use cases

Data in the package relies on the IAM Consortium (IAMC) 'pyam' open-source software data structure (Gidden & Huppmann, 2019) and fits into the IAMC scenario assessment pipeline prepared in support of the IPCC literature assessment.

As part of the pipeline, emissions projections are also harmonised, i.e. modified to be consistent with known historical emissions in a smooth way (Gidden, Fujimori, et al., 2018). The Silicone process is assumed to be part of the IAMC pipeline after harmonisation, as the harmonisation process will potentially differently affect the infilled and infiller data, resulting in inconsistencies. All infiller options except latest time ratio are designed such that if both the data being infilled (the 'infillee data') and the data drawn on for infilling ('infiller data') are harmonised, the result must also be harmonised, so there is no need for harmonisation again after infilling. (Latest time ratio only preserves the harmonisation of the last timepoint in the infiller database.) The infilled results can then be run via climate models, most easily via the OpenSCM package (Nicholls, Gieseke, et al., 2020).

We now demonstrate several uses of the package for specific purposes. The notebooks demonstrating the steps for these calculations can be found in the ”silicone_examples” GitHub repository (Lamboll, 2020). All methods work on databases with only a single region at a time, although the region can be different between the infiller and infillee databases. In our case, we always use global data.

4.2.1. Infilling the Shell Sky scenario

To demonstrate the uses of this package alone, we will apply the methods directly using unharmonised data in the SR1.5 repository (Huppmann et al., 2019; IPCC, 2018) to infill the emission pathways of the Sky scenario by the Shell World Energy Model (which is also found in this database). The Sky scenario only reports a limited set of greenhouse gases and is thus an excellent use case. The crunchers are all used via the multiple infiller, “infill_all_required_emissions_for_openscm”. No active decisions are taken except to use the SSP5 scenarios from the REMIND-MAgPIE 1.5 model for the specified model interpolation. The choice of SSP5 is due to the Sky scenario involving similar assumptions of high-industry, high-development, high energy demand and high fossil fuel use. The choice of model is because this is the marker model for that scenario (Riahi et al., 2017).

We see from Figure 3: Left: The Shell Sky model CO_2 from Industry and Industrial Applications data. The fine lines represent the different timeseries in the SR1.5 database used to perform the infilling and are not included in the legend for clarity. Right: The results of interpolating this data using four different crunchers. The interpolate specified model approach used the REMIND-MAgPIE 1.5 model and only choses scenarios based on SSP5 pathways. that the linear interpolation model (without filtering the database) provides a chaotic pathway, due to its value being determined only by the two points either side of it in the database, which changes randomly with time. Although the interpolate specified model approach is also determined by only a few model/scenario pairs because there is only data from a small number of related scenarios, the pathway is smoother and more consistent. The EQW pathway looks somewhat jagged, due to the density of pathways around the Shell sky model. This means that small differences in CO_2 emissions can result in reasonably large changes in quantile and hence infilled methane emissions. The RMS closest pathway is consistent by construction (and precisely overlines a point in the original database). The quantile rolling windows result also looks consistent and is noticeably closer to the dense cloud of values around 200 Mt CH_4 than other crunchers, which ignore data that is not highly similar in some way. The RMS closest result is more consistent over time but more arbitrary in its selection of the pathway, while quantile rolling window is more conservative in the sense of giving results closer to the median behaviour of the whole data set.

Splitting up a Kyoto Greenhouse Gases path

The silicone package has features that can split a basket of gases into its constituents. In this example we take data from the Climate Action Tracker (CAT) website, which reports projected global emissions in terms of Kyoto gas totals (Jeffery, Guetschow, Gieseke, & Gebel, 2018). While it is possible to use this to infill all other values directly as above, the subcategories of Kyoto gas will not necessarily add up to the Kyoto gas total, so a preferable method is to divide the basket into its constituent parts (CO_2, CH_4, N_2O and F-gases), first using the 'decompose collection' multiple infiller, then infill the remaining gases using other techniques as before. F-gases could be further subdivided using similar methods.

As can be seen in Figure 5, the curves that result from decompose collection are generally smooth, in spite of being separately calculated at each timepoint. It is important to ensure that the number of scenarios reported at each time are consistent. In the SR1.5 database, some scenarios only report values at decadal intervals, whereas others use five-year intervals. We interpolated all models to five-year intervals to give consistent representation. In the CH_4 and F-gases, the lowest orange line is clearly seen to rise discontinuously after 2060. This is the last point before the Kyoto total goes negative. To ensure that the sign of the constituents is correct, the formula only considers data from SR1.5 paths where the Kyoto total has the same sign as in the data being infilled. In this way, emissions that are unlikely to go negative like methane are ensured positive, however their magnitude increases the more negative the aggregate is. An alternative way to infill, which is preferable in this case where we know that only one of the emissions will go negative, is to use RMS closest or (non-ratio) quantile rolling windows to infill the positive values and then allow the value that may be negative (CO_2) to make up the rest. This produces the results seen in Figure 6. Here behaviour of all curves is fairly smooth, with no obvious features around zero-crossing points and no negative values except in CO_2, as expected.

4.2.2. Stylised trajectories

Another use of this software is to infill simple, stylised trajectories generated to explore a wide range of possibilities without detailed economic modelling. For example, Sanderson *et al.* (Sanderson, O'Neill, & Tebaldi, 2016) suggest simple formulae whereby one may construct emissions trajectories characterised by a few free variables – in this case, based on transitioning between the RCP pathways. They present general formulae for generating plausible total CO_2 pathways with several free variables. Silicone provides an alternative means of complementing such results by infilling using scenarios with similarities to the desired narrative, removing the need for more numerical free parameters between formulae. A notebook can be found in the silicone examples github detailing the calculations and demonstrating this usage, titled "Infill_stylised_path.ipynb" (Lamboll, 2020), using data from (Nicholls, Meinshausen, et al., 2020). It shows that curves with different values in some of the parameters, termed E_∞ and τ, can be complemented using a number of techniques. Here we highlight the method of interpolating results from any of the SSP scenarios as implemented by the MESSAGE models. As the different SSP models have different narratives, this allows the user to decide what narrative is relevant to the infilling, rather than adding more arbitrary values (Gidden, Riahi, et al., 2018). An example of this output can be found in Figure 7.

5. Summary

In this paper we have outlined the features of the open-source Silicone package. This provides tools for complementing emissions pathways with other climate-relevant emissions through relationships found in the scenario literature. The package features several scripts for analysing data to establish the relationships between the variables in the complete infiller database, to establish the best variables to use when infilling. The values of the follower data are estimated using objects called crunchers, of which there are many. Notebooks describing the use of the crunchers are included in the GitHub repository, which also contains full documentation, and a flowchart for choosing the best cruncher for your situation is included in the text. The results of Spearman's rank correlations implied that, in cases of complete ignorance, the best guess at lead variables are methane and CO_2, and that the best default cruncher is the root mean squared closest cruncher, followed by the quantile rolling windows cruncher. Both of these perform significantly better at reconstructing known pathways compared to the currently-used equal quantile walk technique, although this and many other crunchers are included in the package for situationally specific usage. Several use-cases of the infilling techniques are demonstrated.

Bibliography

Gidden, M. J., Fujimori, S., van den Berg, M., Klein, D., Smith, S. J., van Vuuren, D. P., & Riahi, K. (2018). A methodology and implementation of automated emissions harmonization for use in Integrated Assessment Models. *Environmental Modelling and Software*, *105*, 187–200. https://doi.org/10.1016/j.envsoft.2018.04.002

Gidden, M. J., & Huppmann, D. (2019). pyam: a Python Package for the Analysis and Visualization of Models of the Interaction of Climate, Human, and Environmental Systems. *The Journal of Open Source Software*. https://doi.org/10.21105/joss.01095

Gidden, M. J., Riahi, K., Smith, S. J., Fujimori, S., Luderer, G., Kriegler, E., … Takahashi, K. (2018). Global emissions pathways under different socioeconomic scenarios for use in CMIP6: a dataset of harmonized emissions trajectories through the end of the century. *Geoscientific Model Development*. https://doi.org/10.5194/gmd-2018-266

Huppmann, D., Kriegler, E., Krey, V., Riahi, K., Rogelj, J., Calvin, K., … Zhang, R. (2019, August 8). IAMC 1.5°C Scenario Explorer and Data hosted by IIASA. https://doi.org/10.5281/zenodo.3363345

Huppmann, D., Rogelj, J., Kriegler, E., Krey, V., & Riahi, K. (2018, December 1). A new scenario resource for integrated 1.5 °C research. *Nature Climate Change*, Vol. 8, pp. 1027–1030. https://doi.org/10.1038/s41558-018-0317-4

IPCC. (2013). *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia,.*

IPCC. (2014a). *Chapter 6, Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Edenhofer, O., R. Pichs-Madruga, Y. Sokona, E. Farahani, S. Kadner, K. Seyboth*.

IPCC. (2014b). *Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)]*.

IPCC. (2018). *Global Warming of 1.5°C.An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change*, . https://doi.org/10.1002/9780470996621.ch50

Jeffery, M. L., Guetschow, J., Gieseke, R., & Gebel, R. (2018). PRIMAP-crf: UNFCCC CRF data in IPCC 2006 categories. *Earth System Science Data Discussions*, 1–21. https://doi.org/10.5194/essd-2018-3

Lamboll, R. D. (2020). Silicone examples github. https://doi.org/10.5281/zenodo.3822152

Lamboll, R. D., Nicholls, Z., & Kikstra, J. (2020a). Silicone documentation. Retrieved May 11, 2020, from https://silicone.readthedocs.io/en/latest/index.html

Lamboll, R. D., Nicholls, Z., & Kikstra, J. (2020b). Silicone github. https://doi.org/10.5281/zenodo.3822259

Meinshausen, M., & Dooley, K. (2019). Mitigation scenarios for non-energy GHG. In *Achieving the Paris Climate Agreement Goals: Global and Regional 100% Renewable Energy Scenarios with Non-Energy GHG Pathways for +1.5C and +2C* (pp. 79–91). https://doi.org/10.1007/978-3-030-05843-2_4

Meinshausen, M., Hare, B., Wigley, T. M. L., Van Vuuren, D., Den Elzen, M. G. J., & Swart, R. (2006, March). Multi-gas emissions pathways to meet climate targets. *Climatic Change*, Vol. 75, pp. 151–194. https://doi.org/10.1007/s10584-005-9013-2

Meinshausen, M., Raper, S. C. B., & Wigley, T. M. L. (2011). Emulating coupled atmosphere-ocean and carbon cycle models with a simpler model, MAGICC6-Part 1: Model description and calibration. *Atmos. Chem. Phys*, *11*, 1417–1456. https://doi.org/10.5194/acp-11-1417-2011

Meinshausen, M., Vogel, E., Nauels, A., Lorbacher, K., Meinshausen, N., Etheridge, D. M., … Weiss, R. (2017). Historical greenhouse gas concentrations for climate modelling (CMIP6). *Geoscientific Model Development*, *10*(5), 2057–2116. https://doi.org/10.5194/gmd-10-2057-2017

Nicholls, Z., Gieseke, R., Lewis, J., & Willner, S. (2020). OpenSCM: Unified access to simple climate models. Retrieved April 6, 2020, from https://github.com/openscm/openscm

Nicholls, Z., Meinshausen, M., Lewis, J., Gieseke, R., Dommenget, D., Dorheim, K., … Xie, Z. (2020). Reduced complexity model intercomparison project phase 1: Protocol, results and initial observations. *Geoscientific Model Development Discussions*, 1–33. https://doi.org/10.5194/gmd-2019-375

O'Neill, B. C., Tebaldi, C., van Vuuren, D. P., Eyring, V., Friedlingstein, P., Hurtt, G., … Sanderson, B. M. (2016). The Scenario Model Intercomparison Project (ScenarioMIP) for CMIP6. *Geoscientific Model Development*, *9*(9), 3461–3482. https://doi.org/10.5194/gmd-9-3461-2016

Riahi, K., van Vuuren, D. P., Kriegler, E., Edmonds, J., O'Neill, B. C., Fujimori, S., … Tavoni, M. (2017). The Shared Socioeconomic Pathways and their energy, land use, and greenhouse gas emissions implications: An overview. *Global Environmental Change*, *42*, 153–168. https://doi.org/10.1016/j.gloenvcha.2016.05.009

Rogelj, J., Popp, A., Calvin, K. V., Luderer, G., Emmerling, J., Gernaat, D., … Tavoni, M. (2018). Scenarios towards limiting global mean temperature increase below 1.5 °c. *Nature Climate Change*, *8*(4), 325–332. https://doi.org/10.1038/s41558-018-0091-3

Sanderson, B. M., O'Neill, B. C., & Tebaldi, C. (2016). What would it take to achieve the Paris temperature targets? *Geophysical Research Letters*, *43*(13), 7133–7142. https://doi.org/10.1002/2016GL069563

Schaeffer, M., Gohar, L., Kriegler, E., Lowe, J., Riahi, K., & van Vuuren, D. (2015). Mid- and long-term climate projections for fragmented and delayed-action scenarios. *Technological Forecasting and Social Change*, *90*(PA), 257–268. https://doi.org/10.1016/j.techfore.2013.09.013

Smith, C. J., Forster, P. M., Allen, M., Leach, N., Millar, R. J., Passerello, G. A., & Regayre, L. A. (2018). FAIR v1.3: A simple emissions-based impulse response and carbon cycle model. *Geoscientific Model Development*, *11*(6), 2273–2297. https://doi.org/10.5194/gmd-11-2273-2018

Weyant, J. (2017). Some contributions of integrated assessment models of global climate change. *Review of Environmental Economics and Policy*, *11*(1), 115–137. https://doi.org/10.1093/reep/rew018

Author contributions

RDL and ZRJN wrote the code, JSK assisted reviewing it. JR and MM conceived infilling techniques. RDL wrote the manuscript, all authors gave comments and contributed to the final version.

Code availability

The Silicone code in this paper is available from the main github repository (Lamboll, Nicholls, & Kikstra, 2020b). Code used to analyse the output of Silicone is available in a second github repository (Lamboll 2020).

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 820829. We thank Nicholai Meinshausen for useful statistics discussions.

385

Table 1: A guide to crunchers. Names followed by asterisks use a ratio-based approach, i.e. they find a multiplicative factor and then multiply the infillee lead by this value. If the asterisk is in brackets, a ratio-based approach is optional. Otherwise, techniques all return linear combinations of values seen in the infiller database.

Name	Description	Use case	Pitfalls
Constant ratio*	Multiplies the lead variable by a constant (not fitted to any data)	Used when no information about the follower variable is available in any database. Mainly used for infilling with zeros.	Has no basis in the data – only used as a last resort in cases of complete uncertainty.
Latest time ratio*	Multiplies the lead variable by a constant fitted to a single (latest) timepoint in the infiller data.	Used when no data is available for most times, this generalises from the latest information we have, e.g. if only historic data is available.	No reason to assume that the relationship between emissions holds for all time. No restriction on signs of follower gas, so potential sign errors when the lead (but not follower) emissions may become negative. Emissions may need re-harmonising.
Time dependent ratio*	Multiplies the lead variable by the ratio of the averages of the lead and follower data in the infiller database. (Note: this ratio is not the same as the average of the ratios and is more stable to inclusion of extreme ratios.) Optionally calculates this using only values with the same sign of lead gas emissions.	Used when two emissions should track each other, or one represents a portion of the other.	Allows arbitrarily high emissions. Can behave unexpectedly if emissions change sign, particularly if emissions with this sign are not seen at the same time in the infiller database.
RMS closest	Finds the most similar pathway in the infiller database and uses those values. Most similar means smallest root mean squared difference between the lead values of infiller pathways and infillee pathway averaged over all times.	Used when behaviour at one time should strongly determine behaviour at another and continuity is needed between times. The only cruncher that does not treat each time separately.	A small change in the input at a single timestep can result in large changes in output at every timestep. All the results returned are found exactly in the infiller database, so if that database is small, the same values are returned in many cases. Results more extreme than found in the infiller database all return the same value.

Linear interpolation / Interpolate specified scenarios and models	At each time, linearly interpolates between the follower values at the two nearest lead values, taking averages where multiple points have identical lead values. Interpolate specified scenarios and models filters the input database before applying the same technique	Used for infilling where we have a small number of comparable models/scenarios.	A small change in the input can result in a large change in the output at the same timestep because pathways in the infiller database can be very different in follower variables for nearly identical values of the lead variable. For similar reasons, results can vary erratically between timesteps for large infiller datasets. Results more extreme than those found in the infiller database all return the same value.
Quantile rolling window (QRW)/time-dependent quantile rolling window (*)	At each time, applies a 1/(1+(lead variable difference)^2) weighting to datapoints at equally spaced points across the infiller lead. Then calculates a specified quantile (usually the median) for the infiller follower value at these points. Can also be used in ratio mode, in which case the ratio between lead and follower in the infiller database is treated as above. Time-dependent QRW allows the quantile to be different at different times (but is computationally slower).	Can choose options to give more smoothing (less noise) or more localised behaviour (shows trends better) Allows the option to generate a distribution of outputs, not just a single optimum. Can add to the narrative through time-dependence. Ratio mode allows better infilling outside the range of the input data.	Using with any quantile larger than 0.5 will result in all emissions being higher, even if the lead and follower emissions anticorrelate. Results more extreme than found in the infiller database all return the same value, unless in ratio mode. In ratio mode, sign changes in the lead variable can result in follower emissions being assigned undesired negative values.
Equal quantile walk (EQW)	Calculates the quantile of the infiller database corresponding to the lead value in each individual year. Returns that quantile in that year of the follow value from the same database.	Conceptually simple, used by previous work.	Assumes all variables are monotonically increasing together. Results more extreme than those found in the infiller database all return the same value.

Table 2: Guide to aggregation tools and multiple infillers.

Name	Description	Use case	Pitfalls
Aggregation tools			
Aggregate to composite values	Requires only the infillee database. Adds together known values to construct a	Infilling aggregate values (e.g. Kyoto gas totals)	Requires all information to be known already – no statistical inference, just adding.

	consistent output (with optional weighting).		
Multiple infillers			
Decompose collection*	Constructs a consistent version of the aggregate in the infiller database. Breaks a known quantity down into components, estimated by the time-dependent ratio method.	Breaking down aggregate values into their components.	Requires all infiller scenarios to have all components, otherwise ratios will be distorted. Any contributions to the aggregate values not quantified in the infiller database are ignored.
Infill all required values (*)	Uses the same lead variable and cruncher to infill any gaps in emissions data.	For infilling scattered, minor gaps in a largely sound database.	Low confidence in the results being accurate as the method does not consider the specific characteristics of the data.

https://doi.org/10.5194/gmd-2020-138
Preprint. Discussion started: 28 May 2020

Table 3. Absolute values of Spearman's Rank correlation between emissions, averaged over the start of decades from 2020 to 2100. We use the following abbreviations: BC as black carbon, VOC as volatile organic compounds, AFOLU as Agriculture, Forestry and Other Land Use; and En & IP as as energy and industrial processes. "CO$_2$|" represents subtypes of CO2. We also calculate the average of these rows, with or without the CO2 and subtypes. Cells are bold if the value in them is > 0.7 and are coloured orange if the variance of the rank correlation between years exceeds 0.03.

variable	BC	CH$_4$	CO	CO$_2$	CO$_2$| AFOLU	CO$_2$|En & IP	F-gases	HFC	N$_2$O	NH$_3$	NOx	OC	PFC	SF$_6$	Sulfur	VOC
BC		0.47	**0.75**	0.46	0.37	0.42	0.23	0.10	0.40	0.40	0.58	**0.73**	0.41	0.20	0.48	0.45
CH$_4$			0.32	**0.74**	0.49	**0.73**	0.64	0.58	**0.86**	0.34	0.58	0.30	0.66	0.41	0.65	0.24
CO				0.36	0.38	0.32	0.06	0.16	0.29	0.35	0.48	**0.78**	0.05	0.17	0.36	0.68
CO$_2$					0.54	**0.96**	0.60	0.57	0.54	0.30	0.61	0.24	0.35	0.22	0.69	0.37
CO2| AFOLU						0.36	0.27	0.40	0.53	0.36	0.33	0.34	0.23	0.21	0.31	0.20
CO2| Energy & IP							0.58	0.51	0.50	0.25	0.61	0.17	0.32	0.18	0.69	0.36
F-gases		🟧				🟧		**0.91**	0.57	0.19	0.50	0.10	**0.90**	**0.77**	0.60	0.12
HFC			🟧		🟧				0.46	0.11	0.30	0.14	**0.71**	0.68	0.36	0.23
N2O										0.44	0.46	0.30	0.65	0.40	0.49	0.17
NH$_3$									🟧		0.23	0.39	0.10	0.05	0.23	0.25
NOx							🟧					0.22	0.53	0.26	**0.76**	0.39
OC													0.20	0.11	0.19	0.41
PFC					🟧	🟧								**0.77**	0.46	0.16
SF$_6$				🟧	🟧	🟧									0.26	0.24
Sulfur												🟧		🟧		0.46
Average	0.43	0.53	0.37	0.50	0.36	0.46	0.47	0.42	0.47	0.27	0.46	0.31	0.43	0.33	0.47	0.32
Average, no CO$_2$	0.43	0.50	0.37	0.46	0.34	0.43	0.47	0.40	0.46	0.26	0.44	0.32	0.47	0.36	0.44	0.32
# scenarios	389	412	353	414	412	414	368	108	411	345	363	363	180	191	412	345

Table 4: Root mean squared error in reconstructing known data using different crunchers, with CO_2 as the lead variable.

Species	QRW	RMS closest	Linear interpolation	EQW
BC	0.129	0.119	0.177	0.161
CH_4	0.086	0.073	0.097	0.094
CO	0.140	0.133	0.182	0.175
F-gases	0.106	0.095	0.125	0.118
HFC	0.152	0.140	0.165	0.159
N_2O	0.110	0.091	0.136	0.134
NH_3	0.175	0.151	0.240	0.240
NOx	0.104	0.095	0.132	0.121
OC	0.144	0.129	0.196	0.202
PFC	0.132	0.101	0.120	0.163
SF_6	0.166	0.126	0.196	0.207
Sulfur	0.094	0.081	0.108	0.104
VOC	0.167	0.144	0.213	0.208
Mean	**0.131**	**0.114**	**0.161**	**0.161**

400

Table 5: Root mean squared error in reconstructing known data using different crunchers, with methane as the lead variable

	QRW	RMS closest	Linear interpolation	EQW
BC	0.128	0.115	0.168	0.155
CO	0.139	0.112	0.177	0.177
CO_2	0.087	0.083	0.104	0.089
F-gases	0.112	0.099	0.132	0.129
HFC	0.163	0.129	0.182	0.178
N_2O	0.078	0.062	0.085	0.075
NH_3	0.170	0.119	0.213	0.239
NOx	0.107	0.099	0.131	0.119
OC	0.142	0.107	0.178	0.180
PFC	0.122	0.066	0.108	0.154
SF_6	0.161	0.127	0.197	0.203
Sulfur	0.096	0.084	0.104	0.106
VOC	0.165	0.126	0.195	0.214
Mean	**0.128**	**0.102**	**0.152**	**0.155**

Figure 1: Flow chart suggesting how to choose the cruncher (peach oblongs) or multiple infiller (yellow oblongs) to use when 405 **infilling.**

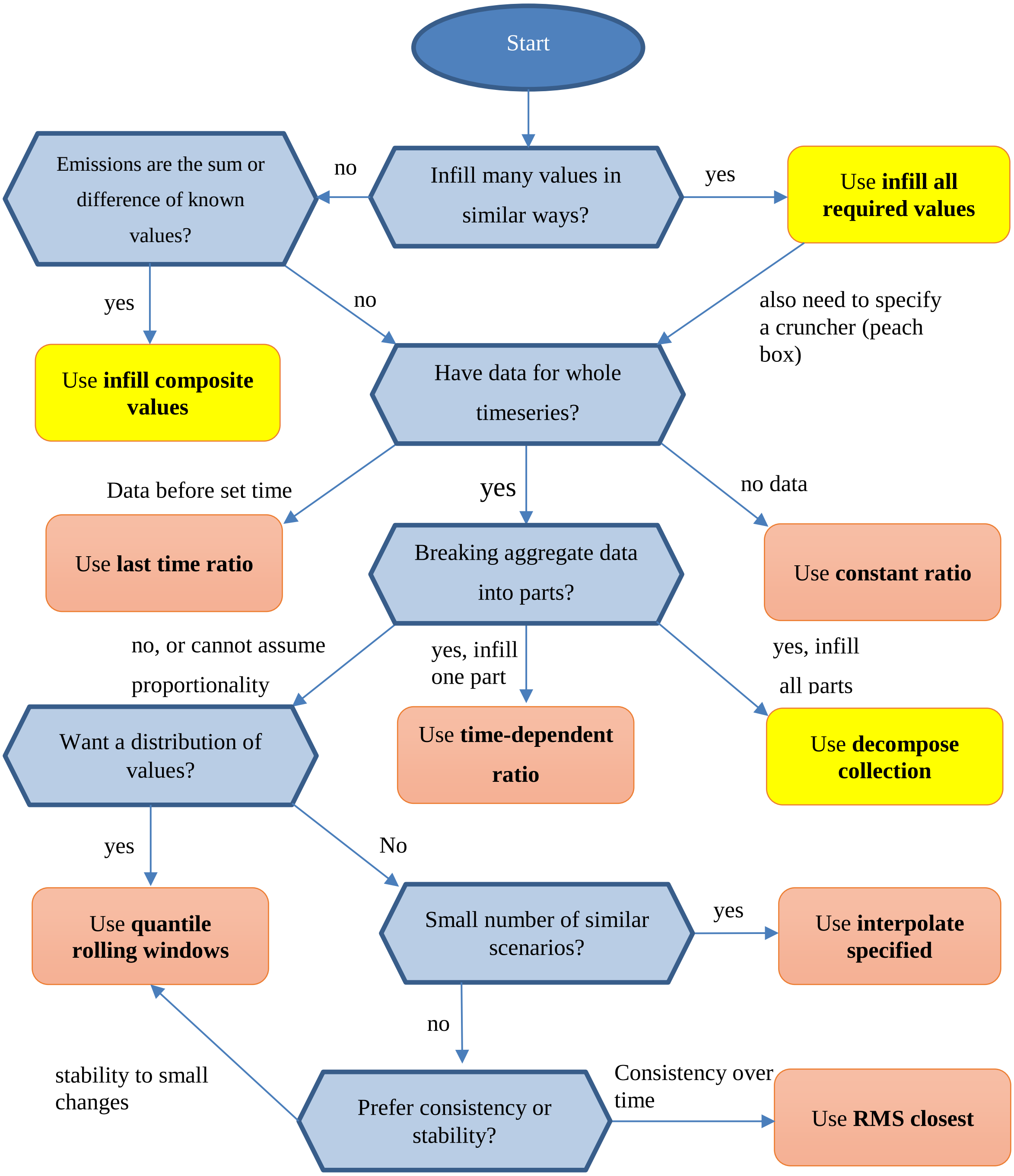

Start
Infill many values in similar ways?
yes
Use **infill all required values**
no
Emissions are the sum or difference of known values?
yes
Use **infill composite values**
no
also need to specify a cruncher (peach box)
Have data for whole timeseries?
Data before set time
Use **last time ratio**
yes
no data
Use **constant ratio**
Breaking aggregate data into parts?
no, or cannot assume proportionality
yes, infill one part
Use **time-dependent ratio**
yes, infill all parts
Use **decompose collection**
Want a distribution of values?
yes
Use **quantile rolling windows**
No
Small number of similar scenarios?
yes
Use **interpolate specified**
no
Prefer consistency or stability?
stability to small changes
Consistency over time
Use **RMS closest**

Figure 2: Schematic of how the quantile rolling window cruncher determines the follow value to use. a) Example relationships between lead (CO_2) and follow (CH_4) variables over time. b) Five windows are drawn and a weighting function constructed for each window. c) A relationship between the sum of the weights and the follow value is established and the follow value at the desired quantile is returned.

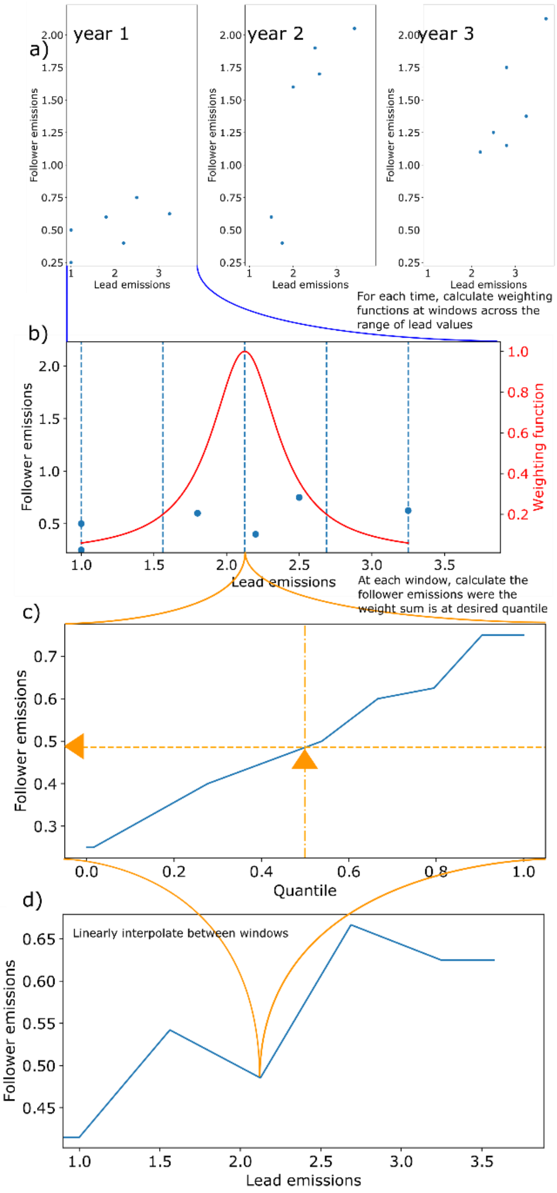


Figure 3: Left: The Shell Sky model CO_2 from Industry and Industrial Applications data. The fine lines represent the different timeseries in the SR1.5 database used to perform the infilling and are not included in the legend for clarity. Right: The results of

415 **interpolating this data using four different crunchers. The interpolate specified model approach used the REMIND-MAgPIE 1.5 model and only choses scenarios based on SSP5 pathways.**

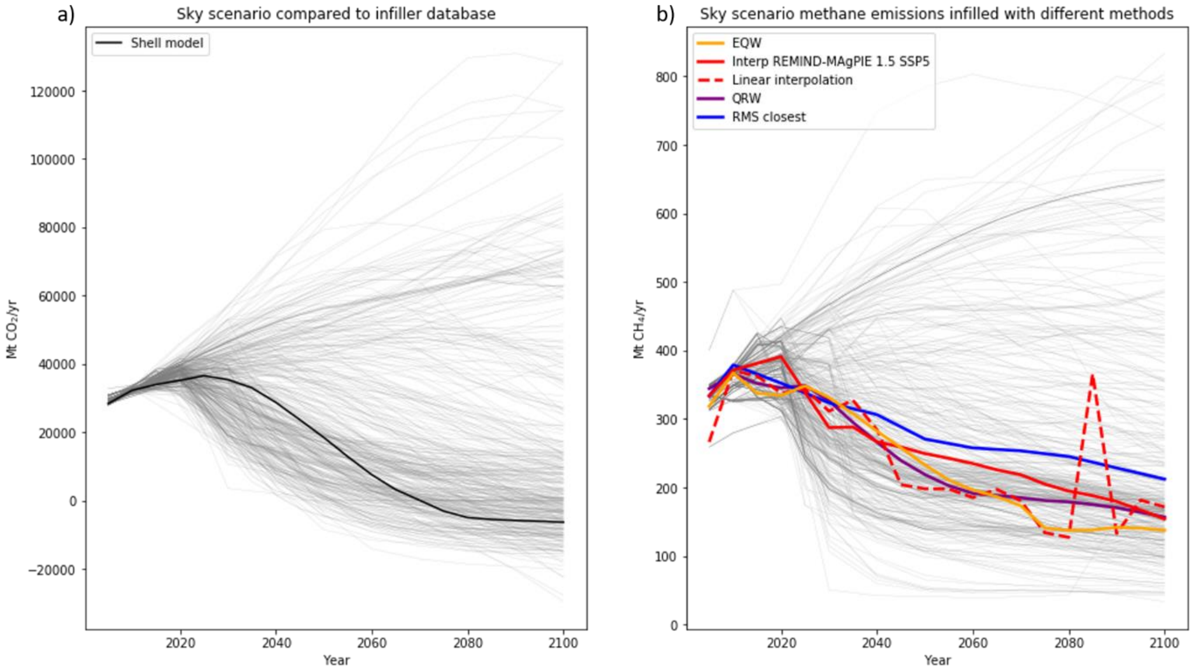


Figure 4: The Climate Action Tracker (CAT) Kyoto gas totals (thick lines) compared with the portfolio of values in the SR1.5 420 **database (thin lines).**

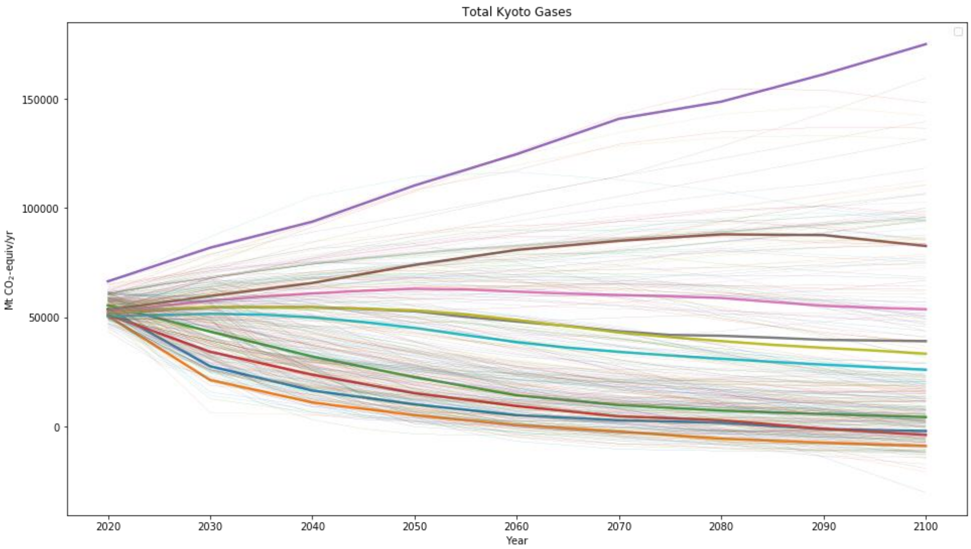
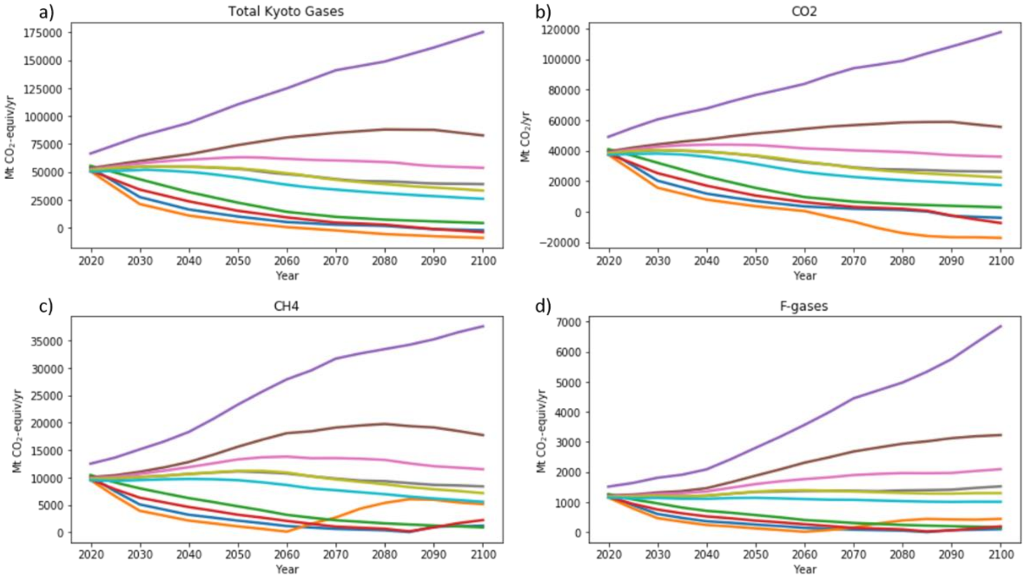


Figure 5: The CAT Kyoto gas baskets decomposed into their components, using the decompose collection multiple infiller.



425

Figure 6: Kyoto gases, decomposed by first infilling the non-negative emissions using the (non-ratio) quantile rolling windows, then infilling the CO$_2$ using infill composite values.

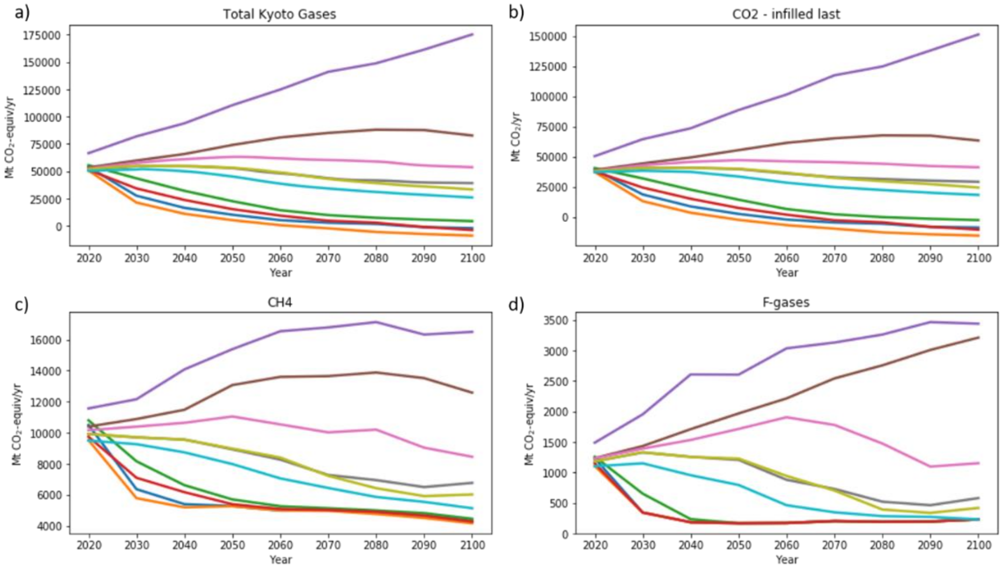
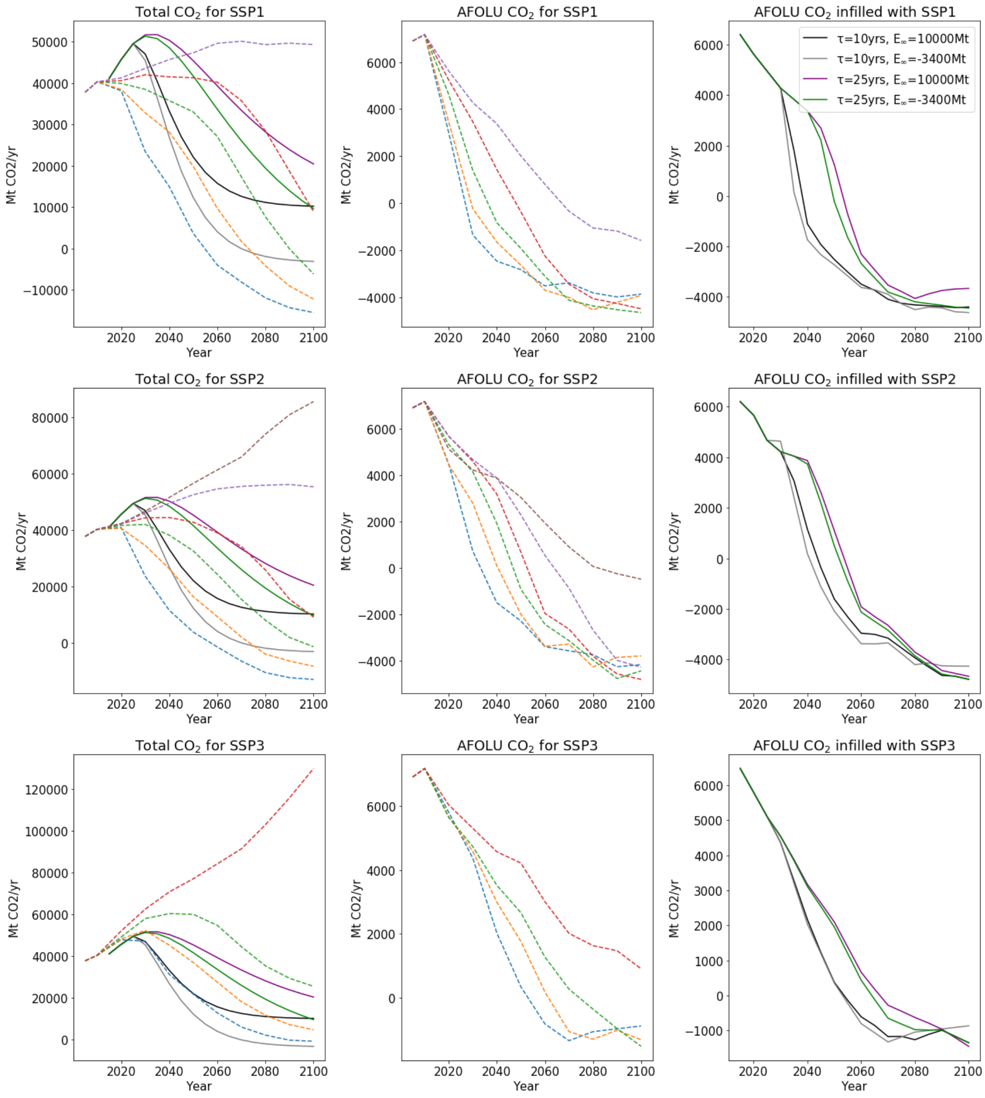


Figure 7: Illustration of using the interpolate specified scenario cruncher to infill a series of stylised trajectories (solid lines), characterised by two different parameters (τ and E_∞), defined in (Sanderson et al., 2016). The first column compares the total CO_2 calculated for the stylised trajectories to the values of the MESSAGE model for a given group of SSP scenarios (dotted lines). These are our lead values in each case. The second column shows the range of follow values for that SSP. The third column shows the resultant AFOLU (Agriculture, Forestry and Other Land Use) trajectories that emerge from using the Interpolate Specified Scenario infiller.


Total CO$_2$ for SSP1
AFOLU CO$_2$ for SSP1
AFOLU CO$_2$ infilled with SSP1
Total CO$_2$ for SSP2
AFOLU CO$_2$ for SSP2
AFOLU CO$_2$ infilled with SSP2
Total CO$_2$ for SSP3
AFOLU CO$_2$ for SSP3
AFOLU CO$_2$ infilled with SSP3
τ=10yrs, E∞=10000Mt
τ=10yrs, E∞=-3400Mt
τ=25yrs, E∞=10000Mt
τ=25yrs, E∞=-3400Mt
Mt CO2/yr
Year