

# Universal, easy access to geotemporal information: FetchClimate

**Dmitry A. Grechka, Sergey B. Berezin, Stephen Emmott, Vassily Lyutsarev, Matthew J. Smith and Drew W. Purves**

*D. A. Grechka and S. B. Berezin, Moscow State Univ., Russia. – S. Emmott, V. Lyutsarev, M. J. Smith (matthew.smith@microsoft.com) and D. W. Purves, Computational Science Laboratory, Microsoft Research, Cambridge, UK. MJS also at: Microsoft Services, Reading, UK.*

Geotemporal information, information associated with geographical space and time, has always been critical to climate and environmental science. However, this information is certainly not universally or easily accessible. In fact, obtaining and using geotemporal information often comes with a considerable technical overheads, impeding research progress. To address this, we introduce FetchClimate: a cloud service designed to provide easy, universal access to geotemporal information. FetchClimate enables and accelerates the use of geotemporal information by enabling it to be accessed programmatically from a Web service (such as the statistical software R) or non-programmatically using a Web browser. We intend the service to accelerate the pace of ecological and environmental research by eliminating the technical overheads currently needed to obtain geotemporal information.

A common practical challenge faced by researchers of ecological and environmental phenomena is obtaining reliable estimates of environmental variables. For example, an ecologist might gather records of species occurrences from multiple locations and then need estimates of the typical climate or soils (Doxford and Freckleton 2012) or might gather weekly data on leaf phenology at a given location over several years, and then need estimates of how air temperature, light and soil moisture varied (Ballestrini et al. 2011). In recent decades, multiple factors have contributed to advances in the availability of environmental data. First, developments in local and remote sensing technology have resulted in many more aspects of the environment being measured. Second, new methodologies for processing environmental data have led to more accurate estimates of environmental variables for locations and times where they have not been measured (Kistler et al. 2001, New et al. 2002, Hijmans et al. 2005). Third, developments in online storage and sharing have resulted in many files containing estimates of environmental variables being posted online (Trabucco and Zomer 2010). Fourth, the drive to ease online access has led to the development of remote data delivery services in which subsets of online data sets can be obtained from a remote server (Kemp et al. 2012, Cinquini et al. 2014).

However, scientists, businesses and society in general still face many challenges in using environmental information and, indeed, any information that is associated with locations in geographical space and time; which we call here ‘geotemporal information’. There are challenges of discoverability and understanding: files containing geotemporal information are scattered across the Web, with few naming standards to help in locating files containing a given

variable (although see Moine et al. 2013 for a description of how naming standards were achieved in a recent Earth system model intercomparison project). Moreover, files vary in their geographical and temporal extents and resolution, the types and uncertainty associated with the estimates, and the legal conditions of data use. As a result, domain knowledge is required of researchers to know which files might be suitable, and where to go to get them. Added to these are multiple technical challenges: usually the files (often large e.g. Gigabytes), need to be downloaded and stored locally, which takes time and storage space. Then, the files must be inspected, filtered, interpolated, and otherwise processed. These technical steps are made harder by the fact that files come in a variety of formats, requiring the researcher to learn to use multiple software tools or libraries to perform analogous tasks on different files. The overall result is that researchers can spend a large fraction of their research time searching for and processing files containing environmental variables, and moreover, do not have the time to experiment or optimize this step to produce the very best estimates possible for them.

Kemp et al. (2012) illustrate one way in which access to geotemporal information can be facilitated, providing a library for the statistical software R to enable users to access from the Web, post-process, and visualise climate variables contained within the NCEP/NCAR reanalysis products. However, that still restricts access to users of R and only provides access to NCEP/NCAR reanalysis products. Many users of geotemporal information will want to access it using other software, including sometimes using techniques that do not involve writing code. They will also typically want access to more information than just the NCEP/NCAR

reanalysis data. We developed FetchClimate to enable more widespread access to geotemporal information and for multiple geotemporal information types to be integrated, whilst eliminating the technical hurdles described above.

## Scope and features

FetchClimate is published as a Web service and, therefore, can be accessed from many software environments, including the statistical software R, or via an interactive Web application (<<http://fetchclimate.org>>). The name FetchClimate originated from the early prototype that only delivered climate information however the service is more general, potentially delivering any geotemporal information. In addition, FetchClimate has been designed to do more data processing steps in the background, such as data set selection and aggregation, targeting a default user experience of simply issuing the fetch for the geotemporal bounds and resolution for a particular variable and receiving results. Making it this simple introduces further challenges such as ensuring reproducibility or selecting the best data selection or processing algorithms to use (matters we address in the Discussion). The FetchClimate server and client libraries are available for download via <<http://fetchclimate.org/about>> and a running Web service is freely available for research use at <<http://fetchclimate.org>>. We have initially populated FetchClimate with a set of commonly used climate, topographic and soil information, as listed and detailed in Table 1, but our longer term aim is for it to be populated with a wider set of research, business and policy critical geotemporal information.

The overarching aim of FetchClimate is to automate the otherwise laborious steps that sit between files posted online and the often limited amount of environmental information that someone needs. We aimed to build a technology where the researcher needs to provide nothing more than a succinct description of the information they need – which variable, for where on Earth, and for which years, days within the year and hours within the day – and the service ‘does the rest’, returning the best available values, together with uncertainty and provenance. The key features of FetchClimate that allow for this behaviour are outlined below, with additional details provided in Supplementary material Appendix 1.

### Automatic data set selection

The researcher is required to specify only the variable they want (e.g. air temperature) alongside the spatial and temporal properties of their query (see below). Given this information, FetchClimate identifies a list of data sets that could fulfil that query, then selects the best data set from among that list. This removes the necessity for specialized knowledge about which files contain the variable of interest, or about which of those files may contain the best estimates. It also obviates the need for the researcher to ever download any individual data set in its entirety. The automatic selection can be overridden in situations that users would prefer to specify the precise data set they wish to query (shown below).

Table 1. Data sets provided by the FetchClimate service. See Table 2 for definitions of the variables provided.

Data set	Provider	Type	Spatial coverage	Temporal coverage	Variables
CRU CL 2.0	Climatic Research Unit, Univ. of East Anglia. (New et al. 2002)	Rectangular grid	Global (excluding Antarctica). 10 arc minute res. grid	Monthly averages for 12 months over years 1961–1990.	airt, prate, dtr, fdf, wdf, reh, sunp wind
WorldClim 1.4 current	UC Berkeley (Hijmans et al. 2005)	Rectangular grid	Global 30 arc seconds res. grid	Monthly averages for 12 months over years 1950–2000.	airt, prate
NCEP/NCAR Reanalysis	NOAA/OAR/ESRL (Kistler et al. 2001)	Rectangular grid	Global 2.5 degree res. grid/T62 Gaussian grid	4-times daily, monthly values, 1948–present	airt, prate
CEM1-BGC CMIP5 rcp85 ETOPO1	NSF/DOE NCAR (Arora et al. 2013) National Geophysical Data Center, NOAA (Amante and Eakin 2009)	Rectangular grid Rectangular grid	Global 0.9 × 1.25 degree res. grid Global 1 arc-minute res. grid	Daily averages, 2006–2100 Current time (no time variation)	airt, prate elev
GTOPO30	USGS, EROS (1996) < <a href="https://ita.cr.usgs.gov/GTOPO30">https://ita.cr.usgs.gov/GTOPO30</a> >	Rectangular grid	Global 30 arc seconds resolution grid	Current time (no time variation)	elev
GHCNv2	NOAA (Peterson and Vose 1997)	Spatial locations	~ 20000 stations on all continents	Monthly averages, 1701–present	airt, prate
CPC Soil Moisture	NOAA/OAR/ESRL PSD (Fan and van den Dool 2004)	Rectangular grid	Global 0.5 degree res. grid	Monthly averages, 1948–present	soilmoist

Table 2. Information types provided by the FetchClimate service.

Variable	Description	Units
airt	Air temperatures near surface	°C
dtr	Diurnal temperature rate	°C
prate	Precipitation rate	mm month <sup>-1</sup>
wind	Wind speed at 10 m	m s <sup>-1</sup>
reh	Relative humidity	%
sunp	Sunshine fraction (percentage of maximum possible sunshine)	%
elev	Land elevation above sea level	m
wdf	Wet day frequency	day month <sup>-1</sup>
fdf	Frost day frequency	day month <sup>-1</sup>
soilmoist	Soil moisture	mm m <sup>-1</sup>

### ***A variety of geotemporal queries***

For any given variable, FetchClimate supports a variety of queries, differing in their treatment of space and time. The researcher specifies a range of latitude and longitude; years; days within the year; and hours within the day. Then, for each of these dimensions (space; years; days; hours) the researcher specifies whether to average over this range, or step through it. Alternatively, for space only, the research can supply a list of locations, rather than average or step through a range (as in a spatial grid). The different combinations of these choices allow for 32 qualitatively different types of query (see Supplementary material Appendix 1, Table 1 for details of all 32 queries). There are queries which are logically sound but which FetchClimate does not currently support (for example, queries over arbitrary geographic polygons, or square grids that do not align in a N-S, E-W direction), but we consider that the set of queries currently supported covers many of the common cases.

### ***Stitching***

The automatic selection of data sets is, by default, ‘fine grained’, in that it is carried out separately for each value required by the query, rather than for the whole query. For example, a query for air temperature over a grid covering both land and sea may end up being fulfilled by using different data sets for those grid-cells over the land surface than for those grid-cells covering the sea. Similarly, different data sets may be used to fulfil different time points in a time series.

### ***Data handlers***

Every dataset added to FetchClimate has to be associated with a data handler. These code libraries, which can apply to individual or multiple datasets, carry out all the intermediate calculations (e.g. interpolation, aggregation) to transform the source data into the geotemporal specifications requested by the user.

### ***Averaging and interpolation***

FetchClimate does not simply filter the data sets to fulfil the query. Rather, it uses averaging and linear interpolation in both space and time in order to provide an estimate for each precise point (or region) specified in the query. For example, if the researcher requests the average air temperature for 1 April, over the years 1990 to 2000, FetchClimate will first use interpolation to calculate the value for 1 April for each year; then average over these values to provide the final result.

### ***Uncertainty calculation***

Where possible FetchClimate calculates the uncertainty on all values. Again this is carried out in a fine-grained manner, separately for each value specified in the query. The uncertainty is then available to the researcher alongside the data fetched. The uncertainty calculation is also key to the automated data selection: the service runs the query against all suitable data sets, then selects the data set that gave the lowest uncertainty (details below). If the data set cannot produce an uncertainty estimate for a particular query then that data set is given the lowest priority.

### ***Provenance***

FetchClimate records which data set was used to fulfil each returned value and includes this information in the fetch result. It does not currently include details of the intermediate calculations conducted on the source datasets in order to produce the fetch result, although this could be laboriously worked out by inspecting the underlying code for the appropriate data handlers. This issue is addressed in the Discussion.

### ***Reproducibility***

Since FetchClimate is primarily intended as a tool for science, it is important that its results are repeatable. A feature of FetchClimate is that it can reproduce every query that it has ever served despite the fact that the available datasets and interpolation methods may have been updated. To achieve this, it records and time-stamps every change to the service configuration (e.g. updating available data, extending dataset time coverage, updating interpolation methods). This information is stored for the whole lifespan of the service. The ability to access the service configuration history enables it to calculate and use the correct service configuration for a specified time moment in the past. If the user wants to reproduce the query made in the past they can supply a timestamp, which is simply a string defining a moment in time. This can be done either in web interface or with programmatic access to the service. Upon receiving the timestamp, the service looks into the configuration history database and serves the query exactly as if it were requested at the time moment specified in timestamp (with the same set of available data and the same set of interpolation methods for this data).

### ***Parallelisation and caching***

FetchClimate is a cloud service that runs over Microsoft’s cloud computing platform Azure. FetchClimate takes advantage of this platform to organize query caching and to run queries in parallel over multiple nodes. This greatly reduces the time taken to fulfil many queries.

### ***Programmatic or Web application access***

FetchClimate can be accessed from any software that can send and receive information from a web service. We include details of how to access the service at <<https://github.com/dgrechka/FetchClimateAPI>>, which also includes example programmatic access using the languages R, PYTHON and JavaScript. One of our intentions with FetchClimate was to lower the complexity of accessing geotemporal information, including removing the need to write computer code.

Therefore we developed the Web application to enable non-programmatic access (<http://fetchclimate.org>), which also facilitates exploration of the information contained in the service. A criticism of non-code based techniques is that reproducibility can be reduced if the user-selected options are not captured (Joppa et al. 2013, Ahmed et al. 2015). To help avoid this we ensured that the user selected details behind the fetch are included in the URL, so that if that URL is accessed by others it will return to them the same fetch. The Web application has been designed with exploratory use and occasional data downloads in mind. At present, the data downloads do not include details of the options chosen to generate the result. As a result, we recommend programmatic access for those wanting to conduct batch fetches or ensure reproducibility.

## Underlying algorithms

Here we provide a mathematical specification of the default service that FetchClimate provides (various default assumptions can be overridden). Such details could be adopted by any geotemporal data delivery service. More details of the software and architecture used in the specific case of FetchClimate are given in the Supplementary material Appendix 1.

FetchClimate users want to fetch data for a variable  $v$  at a specific set of latitude, longitude and times  $X = \{X_i = (lat_i, lon_i, time_i)\}$ , where we use  $\{\}$  to indicate a set (in this case of multiple latitude, longitude and time combinations) and  $i$  to indicate a member of the set. Users can also fetch data for multiple variables via the Web browser application but here we will focus on the fetching of a single variable.

The data held in FetchClimate updates over time and so to ensure reproducibility FetchClimate users can also specify a timestamp  $\Gamma$ , for the fetch, which tells the service to obtain the results from the version of the FetchClimate data that was available at a particular date. Given these specifications (when  $\Gamma$  is omitted FetchClimate just returns the most up to date data) the service returns

$$Y = F(X, v, \Gamma),$$

where the set of returned data  $Y$ , is uniquely determined by a function  $F$ , of the user specified  $X$ ,  $v$  and  $\Gamma$ .

The function  $F$  is complicated because FetchClimate fulfils the request by consulting a set of available data sets  $\{D\}$ , where  $D$  is an individual data set, which themselves can contain data on multiple variables. FetchClimate finds the data sets that were available at timestamp  $\Gamma$  and that can provide values for  $v$  for each  $X_i$ . It then applies data set specific functions  $G$  (encoded in the data handlers) to each data set at each  $X_i$  to derive values and their uncertainty. Finally, a data set selection function is run over these results to select the value with the lowest uncertainty from among multiple candidate solutions. After this procedure the set of data returned to the user is

$$Y = \{Y_i = (V_i, U_i, P_i)\}$$

where  $V_i$  is the value of  $v$  at  $X_i$ ,  $U_i$  is uncertainty of that value and  $P_i$  identifies the provenance of that particular value. This highlights that FetchClimate can return a data set that has data originating from multiple different sources.

The functions,  $G$  in the data handlers transform data values in the original format in which they were uploaded into FetchClimate (e.g. gridded air temperature at a particular resolution) into values in a potentially different space and time resolution (e.g. through Kriging; Krige 1951), accounting for both the data values and their uncertainty (details in Supplementary material Appendix 1). For instance, if the original data consists of spatially scattered observations (e.g. weather station data) and the user requests the area where the station density is poor, then the interpolated values will be more uncertain than the values for the area where the station density is high. For each space and time point in the request, FetchClimate records data sets that give a real value from computing  $G$  (which is then operated on by the data set selection algorithm).

## Example fetches from R

Fetchclimate can be accessed either using a .NET library (available for download via. <http://fetchclimate.org/about>) or through any other software that can issue commands to a Web service and handle the returned data. We provide examples here using R because of its popularity in ecological and environmental research. We also provide examples for the languages Python and JavaScript at <https://github.com/dgrechka/FetchClimateAPI>.

We wrote the package Rfc to facilitate accessing FetchClimate from R (details at <https://cran.r-project.org/web/packages/Rfc/index.html>; source code at <https://github.com/dgrechka/Rfc>). At time of writing this package contains functions for a limited variety of fetches, as summarized in 'A list of end-user functions' in Supplementary material Appendix 1. The package uses three external code libraries: httr, a network client interface for R (Wickham 2016; <https://cran.r-project.org/web/packages/httr/index.html>), jsonlite, allowing conversion of R objects to and from Javascript object notation (JSON) format (Ooms 2014; <https://cran.r-project.org/web/packages/jsonlite/index.html>) and sp, containing classes and methods for spatial data (Pebesma and Bivand 2005, Bivand et al. 2013; <http://cran.r-project.org/web/packages/sp/index.html>). To run Rfc it first needs to be installed on the host computer (e.g. `install.packages("Rfc")`) and then loaded for the active session (e.g. `library(Rfc)`).

### Fetching a single time series for a point and a range of times

Our first example illustrates fetching yearly time series of summer air temperature values for a single arbitrarily chosen location in Nigeria, Africa.

```
ts <- fcTimeSeriesYearly(variable = "airt",
  latitude = 8.0, longitude = 10.0,
  firstDay = 152, lastDay = 243,
  firstYear = 1950, lastYear = 2050,
  reproduceFor = '2015-05-27')
```

where `ts` is a collection of objects returned from the fetch (details below), 'airt' denotes that the variable of interest

is air temperature, latitude and longitude denote the location for which data is being fetched, firstYear and last year denote the bounds in time for the fetch and reproduceFor defines the date for which the service configuration must be reproduced. The time series list object contains the times for which the fetch was requested, the returned time series data, an estimate of the standard deviation about the returned data values and the provenance of the data values: these can be accessed using the years, values, sd and provenance fields, respectively. The time series that results from this fetch (Fig. 1; plotting commands given in Supporting Information) shows clear effect from using different time series to inform the fetch which would need to be corrected before using the data in most applications. We show how to do this below by fetching data with a specific provenance.

### Fetching a 2D grid for a fixed time

Our second example illustrates the fetching of a variable for a rectangular spatial grid covering all of the African continent

```
africaJulyTemp <- fcGrid(variable = "airt",
  latitudeFrom = -35, latitudeTo = 35, latitudeBy = 1,
  longitudeFrom = -20, longitudeTo = 60, longitudeBy = 1,
  firstDay = 182, lastDay = 212, #July
  firstYear = 1950, lastYear = 2000,
  reproduceFor = '2015-05-27')
```

where africaJulyTemp is a collection of objects returned from the fetch, the latitude and longitude terms define the bounds and resolution of the spatial grid requested and the firstDay and lastDay terms specify the days over which to fetch (other variables defined as before). The returned africaJulyTemp object contains similar data to the previous

fetch but also includes grid coordinates, bounding box and details of the returned projection. Given this data it is then straightforward to generate a map of temperature (Fig. 2a; R commands given in Supplementary material Appendix 1), data provenance (Fig. 2b) and uncertainty (Fig. 2c). Note that for this particular fetch FetchClimate combines two different data sets for land and sea, each with different uncertainties associated with them.

### Overriding defaults: specifying the data set

The automatic selection of data sets can be overridden by specifying the dataSet parameter in the fetch command. For example, a user might choose to only use data from the GHCNV2 data set in the fetch that led to Fig. 1, given that there is a clear effect of the data provider used. This is done by including 'dataSet = "GHCNV2"' in the options list for the fcTimeSeriesYearly command above (Fig. 3; plotting commands given in Supplementary material Appendix 1).

### Discussion

With FetchClimate we sought to simplify access to geotemporal information, although we acknowledge that this can come at the cost of reducing reproducibility or interpreting the fetches incorrectly (Joppa et al. 2013, Ahmed et al. 2015). While the Web application includes details of the user selected options in the URL, reproducibility could be increased further by ensuring such details are also included in any downloaded data. Another useful future development of the web application would be for it to generate code snippets for the fetches to help the user progress to programmatic access; which would help if they want to move to batch or on-demand fetches. A general improvement to

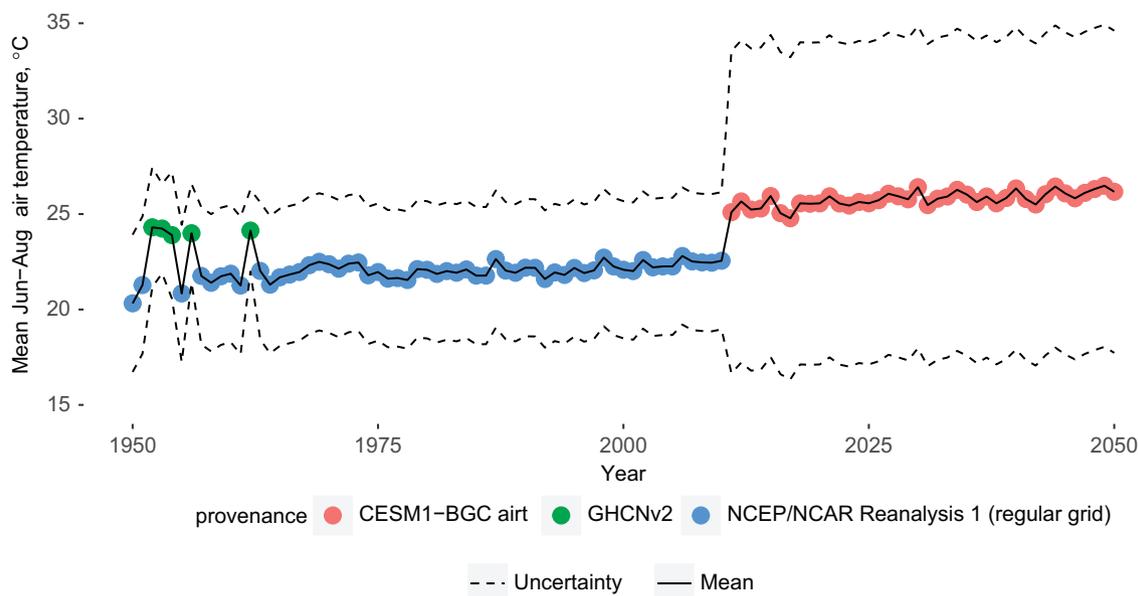


Figure 1. Yearly timeseries for a single geographical location in Nigeria, Africa with location indicated in Fig. 2. Provenance denotes the data sets used to fulfil the query with the data set code defined in Table 1. See the main text for the R code used to fetch this data. Uncertainty indicates one square root of Kriging variance about the mean.

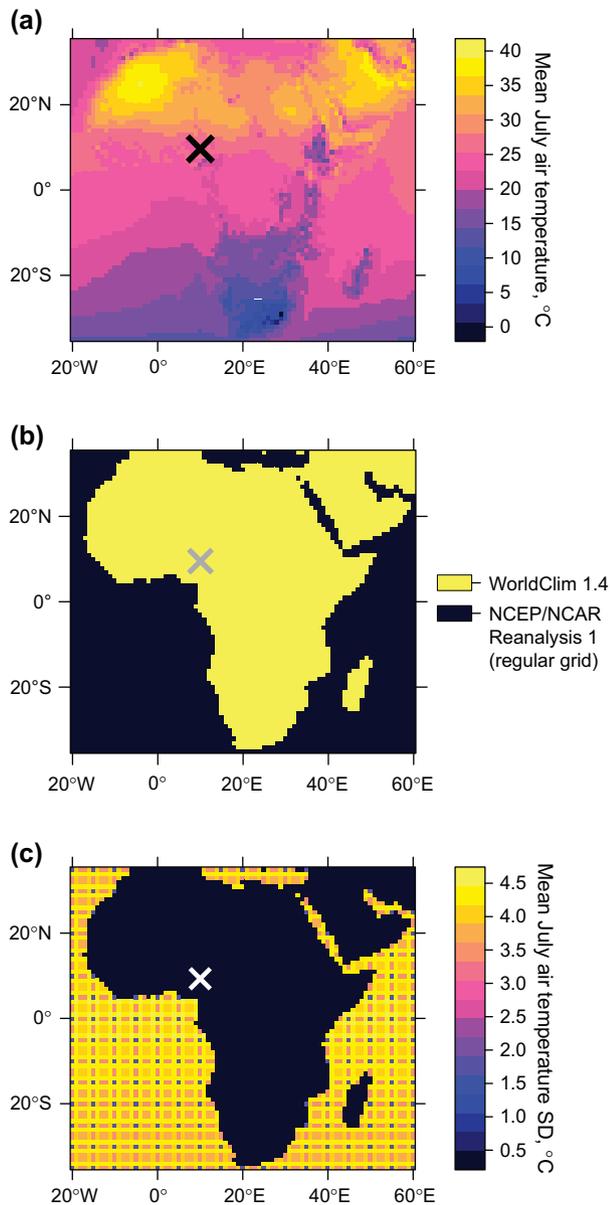


Figure 2. (a) Average July air temperature for the African continent and surrounding seas. Cross indicates the location used for the data set in Fig. 1. See the main text for the R code used to fetch this data. (b) Provenance of the data sets used in the response to the fetch shown in (a). Provenance denotes the data sets used to fulfil the query with the data set code defined in Table 1. Cross indicates the location used for the data set in Fig. 1. See the main text for the R code used to fetch this data. (c) Uncertainty estimates for the data sets used in the response to the fetch shown in (a), where uncertainty is one square root of Kriging variance. Cross indicates the location used for the data set in Fig. 1. See the main text for the R code used to fetch this data. The checkerboard pattern of uncertainty for the ocean occurs due to the Kriging based method of uncertainty estimation which is a function of uncertainty that has been estimated for the regular grid points in the source dataset and the distance from these points (see Supplementary material Appendix 1, ‘Uncertainty evaluation’ section for details of how we perform uncertainty estimation).

the reproducibility of fetches returned by FetchClimate would be to enable it to return details of the intermediate

calculations performed on the source datasets to fulfil a request. It is currently only possible to work this out from rather laborious inspection of the FetchClimate data handler.

The data handlers in FetchClimate use relatively simple algorithms to interpolate and aggregate (details in Supplementary material Appendix 1). However statistical interpolation and aggregation over geotemporal data is itself a large area of research. For example, there are a multitude of options for conducting climatological downscaling including whether to use purely empirical and statistical approaches, whether to downscale using simulation models (dynamical downscaling), and whether it is important to account for the covariation between variables (Harris et al. 2014, Ekström et al. 2015). In many circumstances, alternative techniques may prove better than the ones currently built into FetchClimate, may be preferred by expert users, and will certainly be developed and refined over time. In particular, the appropriateness of our technique to select the best dataset amongst alternatives by evaluating uncertainty has not been assessed in different applications and should be in future: in current form it can sometimes lead to unwanted discontinuities in the returned datasets when the results from different sources have been stitched together (as is apparent in Fig. 3). Natural future developments for FetchClimate will therefore be to advise on the appropriate selection of algorithm in the service, or to enable advanced users to choose the downscaling algorithm for themselves.

Fetchclimate includes a basic command line tool to enable the uploading of new datasets. Adding a new dataset involves uploading the dataset, associating variables in the dataset with FetchClimate information types (potentially requiring new information types to be added), adding metadata, and associating an appropriate data handler with the uploaded data (potentially requiring the creation of a new data handler). Further details on this functionality are given at <<http://fetchclimate.org/about>>. At present, users wishing to do add data will need to set up their own FetchClimate instance; details at <<http://fetchclimate.org/about>>. The types of data for uploading to FetchClimate can be anything that can be represented as a regular grid in geographical space and time. The source data itself can be point or gridded data but the data handler needs to be able to compute over that data and return an answer that is gridded (with single time series, points etc. being a trivial simplification of this).

## Conclusion

The key benefits users gain from FetchClimate are the abilities to inspect and access geotemporal information easily and rapidly. We hope that this service will encourage and enable more widespread use of such information in ecological and environmental research, business and policy. At present, FetchClimate provides information on a few predominantly climatic variables (which inspired its name). Clearly the service could also provide information on any data set gridded in space and time. The extension of such a service to more physical and biological variables, and those that span the past (distant to near term) and the future, could greatly expand the numbers of uses and users of such a service.

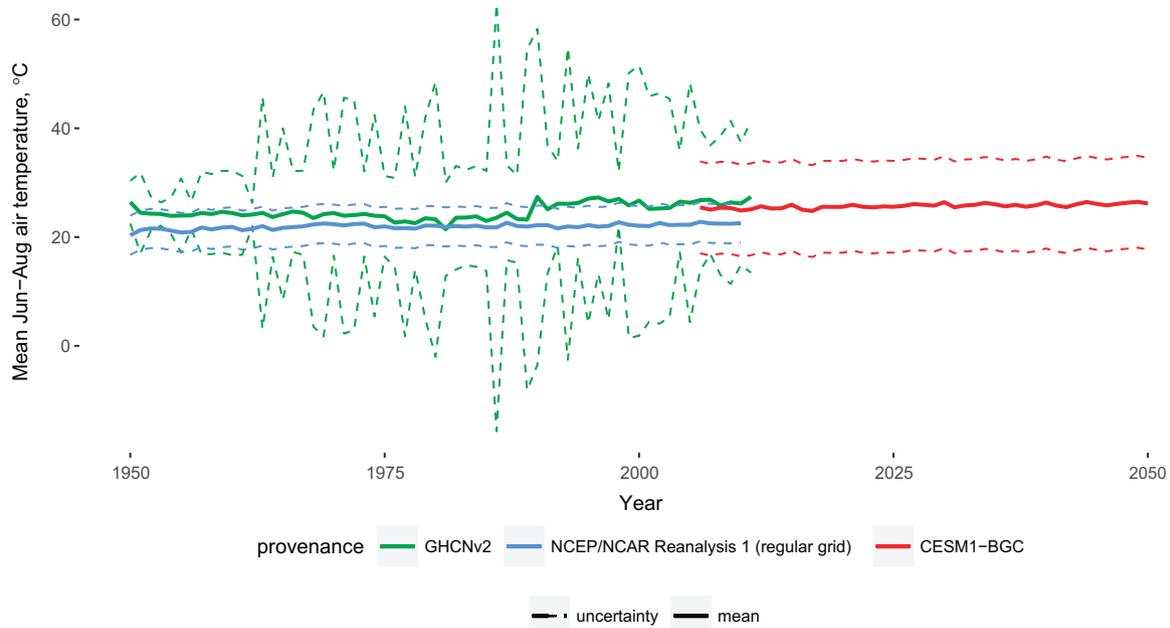


Figure 3. Same fetch as shown in Fig. 1 but with individual data sets selected. See legend for Fig. 1 for definitions.

To cite FetchClimate or acknowledge its use, cite this Software note as follows, substituting the version of the application that you used for ‘version 0’:

Grechka, D. A., Berezin, S. B., Emmott, S., Lyutsarev, V., Smith, M. J. and Purves, D. W. 2016. Universal, easy access to geotemporal information: FetchClimate. – *Ecography* 39: 000–000 (ver. 0).

*Acknowledgements* – We thank Microsoft Research Outreach for enabling the development of FetchClimate, especially Kenji Takeda and Kristin Tolle for their helpful advice, discussions and support. FetchClimate is supported by a Microsoft Azure for Research grant to Moscow State Univ. We thank Sadia Ahmed and Silvia Caldararu for advice on the FetchClimate design. We thank Dmitry Voytsekhovskiy for system architecture discussions. We acknowledge the World Climate Research Programme’s Working Group on Coupled Modelling, which is responsible for CMIP (Arora et al. 2013), and we thank the climate modeling groups for producing and making available their model output. For CMIP the U.S. Department of Energy’s Program for Climate Model Diagnosis and Intercomparison provides coordinating support and led development of software infrastructure in partnership with the Global Organization for Earth System Science Portals. NCEP Reanalysis data was provided by the NOAA/OAR/ESRL PSD, Boulder, Colorado, USA, from their Web site at <www.esrl.noaa.gov/psd/>.

## References

Ahmed, S. E. et al. 2015. Scientists and software-surveying the species distribution modelling community. – *Divers. Distrib.* 21: 258–267.

Amante, C. and Eakin, B. W. 2009. ETOPO1 1 arc-minute global relief model: procedures, data sources and analysis. – NOAA Technical Memorandum NESDIS NGDC-24.

Arora, V. K. et al. 2013. Carbon-concentration and carbon-climate feedbacks in CMIP5 Earth system models. – *J. Clim.* 26: 5289–5314.

Ballestrini, C. et al. 2011. Environmental drivers of leaf phenology in trees of the tropical species *Ficus obtusifolia*. – *Braz. J. Plant Physiol.* 23: 113–122.

Bivand, R. S. et al. 2013. Applied spatial data analysis with R, 2nd ed. – Springer, <www.asdar-book.org/>.

Cinquini, L. et al. 2014. The Earth System Grid Federation: an open infrastructure for access to distributed geospatial data. – *Future Generation Comput. Syst.* 36: 400–417.

Doxford, S. W. and Freckleton, R. P. 2012. Changes in the large-scale distribution of plants: extinction, colonisation and the effects of climate. – *J. Ecol.* 100: 519–529.

Fan, Y. and van den Dool, H. 2004. Climate Prediction Center global monthly soil moisture data set at 0.5 degree resolution for 1948 to present. – *J. Geophys. Res.* 109: D10102.

Ekström, M. et al. 2015. An appraisal of downscaling methods used in climate change research. – *WIREs Clim. Change* 6: 301–319.

Harris, R. M. B. et al. 2014. Climate projections for ecologists. – *WIREs Clim. Change* 5: 621–637.

Hijmans, R. J. et al. 2005. Very high resolution interpolated climate surfaces for global land areas. – *Int. J. Clim.* 25: 1965–1978.

Joppa, L. N. et al. 2013. Troubling trends in scientific software use. – *Science* 340: 814–815.

Kemp, M. U. et al. 2012. RNCER: global weather and climate data at your fingertips. – *Methods Ecol. Evol.* 3: 65–70.

Kistler, R. et al. 2001. The NCEP-NCAR 50-year reanalysis: monthly means CD-ROM and documentation. – *Bull. Am. Meteorol. Soc.* 82: 247–267.

Krige, D. G. 1951. A statistical approach to some mine valuations and allied problems at the Witwatersrand. – Master’s thesis of Univ. of Witwatersrand.

Moine, M. P. et al. 2013. Development and exploitation of a controlled vocabulary in support of climate modelling. – *Geosci. Model Develop. Discussions* 6: 2967–3001.

New, M. et al. 2002. A high-resolution data set of surface climate over global land areas. – *Clim. Res.* 21: 1–25

- Ooms, J. 2014. The jsonlite package: a practical and consistent mapping between JSON data and R objects. – arXiv: 1403.2805 [stat.CO], <<http://arxiv.org/abs/1403.2805>>.
- Pebesma, E. J. and Bivand, R. S. 2005. Classes and methods for spatial data in R. – R News 5, <<http://cran.r-project.org/doc/Rnews/>>.
- Peterson, T. C. and Vose, R. S. 1997. An overview of the Global Historical Climatology Network temperature database. – Bull. Am. Meteorol. Soc. 78: 2837–2849.
- Trabucco, A. and Zomer, R. J. 2010. Global Soil Water Balance Geospatial Database. – CGIAR Consortium for Spatial Information. Published online, available from the CGIAR-CSI GeoPortal at: <[www.cgiar-csi.org](http://www.cgiar-csi.org)>.
- USGS, EROS 1996. Global 30 Arc-Second Elevation GTOPO30. – <<https://lta.cr.usgs.gov/GTOPO30>> accessed 5 October 2015.
- Wickham, H. 2016. httr: tools for working with URLs and HTTP. – <<https://cran.r-project.org/web/packages/httr/index.html>>.

Supplementary material (Appendix ECOG-02321 at <[www.ecography.org/appendix/ecog-02321](http://www.ecography.org/appendix/ecog-02321)>). Appendix 1.