dsclim: A software package to downscale climate scenarios at regional scale using a weather-typing based statistical methodology

Christian Pagé
Laurent Terray
Julien Boé

Climate Modelling and Global Change
TR/CMGC/09/21

Cerfacs
Centre Européen de Recherche
et de Formation Avancée en Calcul Scientifique
42, avenue G. Coriolis, 31057 Toulouse Cedex, France

September 6, 2010

Abstract

Nowadays, global climate scenarios are still provided at coarse spatial and temporal resolution because of computing power limitations. This coarse resolution is generally insufficient to provide adequate data to the climate impacts community at regional scales. To overcome these limitations, several techniques have been developed to downscale climate scenarios. These techniques are either dynamical or statistical based. Dynamical downscaling techniques are still expensive and require a significant amount of computing power, while statistical ones don't require much. These low requirements of statistical downscaling techniques makes them suitable for uncertainty studies because it is thus possible to downscale a large number of different climate scenarios.

An innovative statistical downscaling methodology based on weather-typing have been recently developed and is described in Boé (2007); Boé and Terray (2008a,b); Boé et al. (2006). This methodology has already been used to provide downscaled climate scenarios over France for many groups in the climate impacts community. Consequently, there is a great interest in this type of methodology. To fulfill these needs, an effort has been done at Cerfacs to implement this methodology in an easier-to-use and configurable software package. This report presents the first version of the dsclim software package documentation along with scientific informations about the methodology implementation.

Keywords: scenario, climate, ipcc, downscaling, regional, scale, statistical, methodology, weather-typing, france, SAFRAN, SCRATCH08, tool, software

Introduction

Nowadays, global climate scenarios are still provided at coarse spatial and temporal resolution because of computing power limitations. This coarse resolution is generally insufficient to provide adequate data to the climate impacts community at regional scales. To overcome these limitations, several techniques have been developed to downscale climate scenarios. These techniques are either dynamical or statistical based. Dynamical downscaling techniques are still expensive and require a significant amount of computing power since they require to run a mesoscale numerical model. Statistical techniques however don't require much computing power since they rely on simple regression analyses between large-scale atmospheric fields and regional-scale fields. These low requirements of statistical downscaling techniques makes them suitable for uncertainty studies because it is possible to downscale a large number of different climate scenarios.

An innovative statistical downscaling methodology based on weather-typing have been recently developed and is described in Boé (2007); Boé and Terray (2008a,b); Boé et al. (2006). This methodology has already been used to provide downscaled climate scenarios over France for many groups in the climate impacts community. Consequently, there is a great interest in this type of methodology. To fulfill these needs, an effort has been done at Cerfacs to implement this methodology in an easier-to-use and configurable software package. This report presents the first version of the dsclim software package documentation along with scientific informations about the methodology implementation. This software is being released with an open-source license under CeCILL (CEA - CNRS - INRIA, 2008).

The documentation is structured in the following layout. First, chapter 1 presents the scientific methodology implementation along with current limitations. Chapter 2 presents the general structure of the algorithm, followed by chapter 3 which presents the most important basic building blocks that have been developed. The software being highly configurable, the whole chapter 4 is devoted to present the configuration parameters which are available. Chapter 5 presents the technical aspects to compile and install the software. Finally, two examples are shown in chapter 6.

1 Methodology implementation and generalities

The spatial and temporal characteristic scales of small-scale processes differ significantly from those resolved by global climate models. Consequently, several methodologies have been developed to perform this scale transfer. These methodologies are either dynamical or statistical based. Dynamical methodologies simulate the full physics and dynamics of the processes at high resolution, but these are very expensive in terms of CPU and disk space. To overcome these limitations, statistical downscaling methodologies have been developed. Some of these are based on the concept of weather regimes, which will be discussed in the next subsection.

Before going into the description of weather regimes, it is necessary to introduce the basics of Boe08 methodology using the following general schema (Fig. 1). This schema shows that a statistical model is built to link the large-scale circulation (predictor variables) and the local-scale climate variables. Consequently, there must exists a link between the two scales. The idea in this context is to define groups of days exhibiting similar large-scale atmospheric circulations which are the most discriminating regarding a local climatic variable of interest (over a specific region). This implies that each of these large-scale circulations (LSC) relates to anomalies of at least one local-scale variable. Knowing the weather circulations over the mid- and high-latitudes, it is likely that these LSC are not all the same over the whole calendar year: they should then be dependent on the seasons.

Large Scale
Climatic State
(predictors)
Statistical
Model
Local Climatic State

FIG. 1: General schema of statistical downscaling methodologies.

1.1 Weather Regimes and Weather Types

The concept of weather regimes have been introduced first during the 1950's in synoptic climatology. It is based on a conceptual representation of atmospheric dynamics of mid- and high-latitudes. A weather regime is characterised by a recurrent weather pattern over a specific region, highly linked with significative weather. Vautard (1990) have defined and analysed the most common weather regimes over the extratropical North Atlantic and Europe (Fig. 2). He shows that there are four recurrent weather regimes patterns.

In the context of the statistical downscaling methodologies based on LSC, the term weather types have been introduced. It differs from the weather regimes in the sense that they are defined as the LSC which are the most discriminating regarding local climatic variables of interest, over a specific region and season. Consequently, they are more adapted for statistical downscaling.

In the case of Boe08's methodology, weather types have been defined over parts of Western Europe (Fig. 2), using the Mean Sea-Level Pressure as the LSC variable. Now, let's examine the seasonal climatologies precipitation and associated

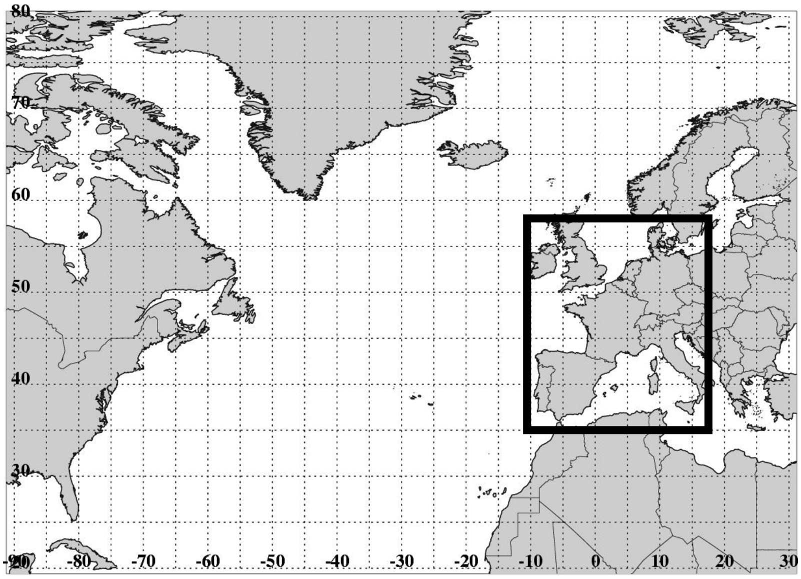

FIG. 2: Geographical domain used by Vautard (1990) to define extratropical North Atlantic and Europe weather regimes. The region used by Boé and Terray (2008a) to define France Weather Types for precipitation is highlighted in black.

Mean Sea-Level Pressure (MSLP) patterns in Figs 3 and 4 to further discuss on the weather typing approach. December-January-February (DJF) months show a wide minimum of MSLP centered over northern Western Europe, while intermediate seasons September-October-November (SON) and March-April-May (MAM) show a much weaker minimum elongated from Western to Eastern northern Europe. Summer (June-July-August (JJA)) shows a strong anticyclone centered over Eastern Mediterranean Sea and encompassing all southern Europe. Precipitations associated with these climatologies exhibit the following characteristics. During Autumn, it is quite wet for almost all of France, especially over the mountains, the Atlantic coast and northern France. During Winter, it is drier than Autumn for regions exposed to the Mistral. During Spring, all the Mediterranean coast is drier, along with plains in northwestern France. Finally, during Summer, the latter regions are even much drier, along with Corsica. Data comes from the NCEP re-analysis (Kistler et al., 2001) for the MSLP field, and from the Météo-France SAFRAN mesoscale analysis (Le Moigne, 2002) for the precipitation observations. SAFRAN is a mesoscale analysis on a regular grid at a resolution of 8-km using a Lambert Conformal projection, and it provides the following hourly variables: 2-meter temperature, 2-meter specific humidity, surface incoming infra-red radiation, surface shortwave incoming radiation, surface liquid precipitation, surface solid precipitation, and 10-meter wind speed.

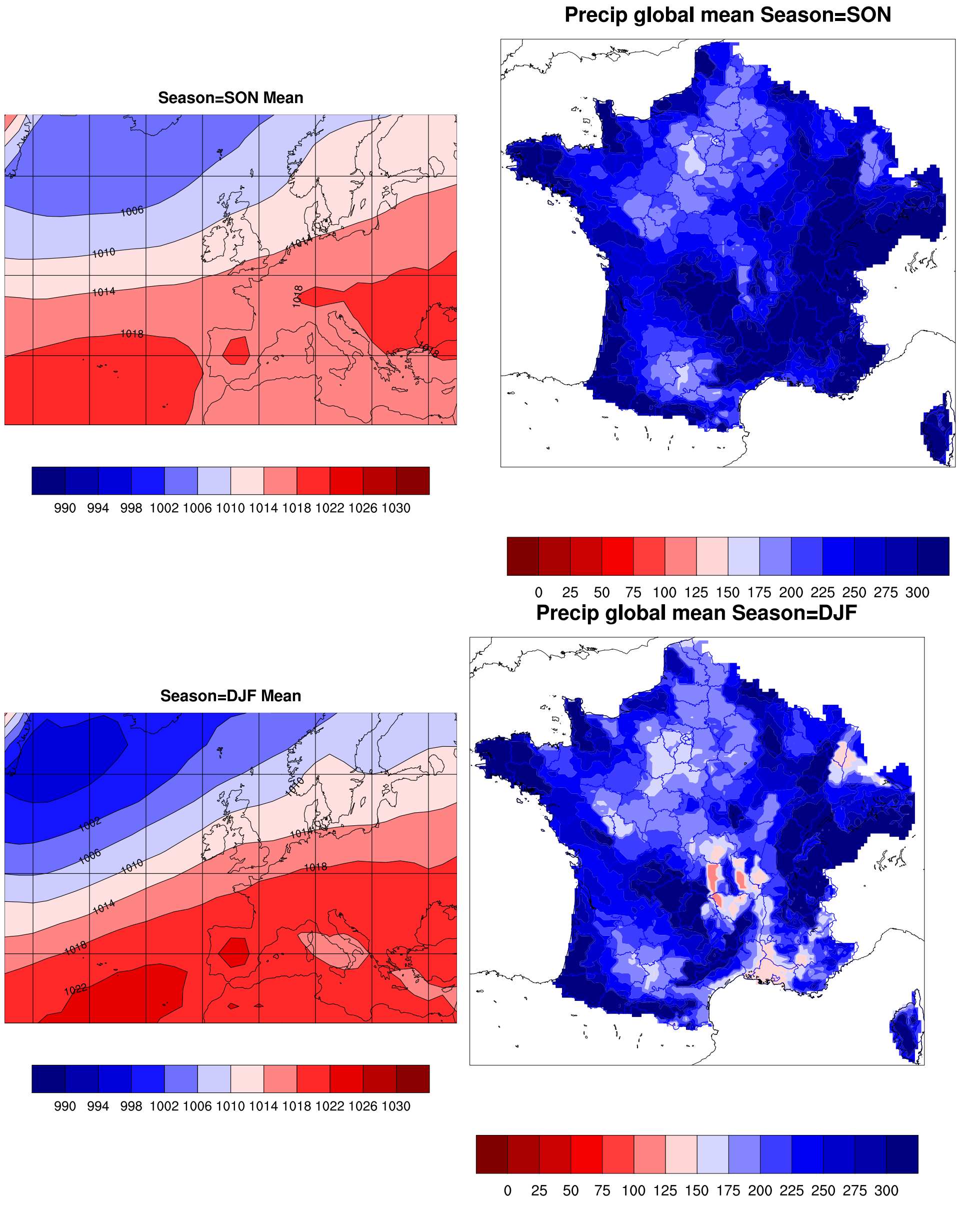

FIG. 3: Seasonal means over Autumn (Sept-Oct-Nov) and Winter (Dec-Jan-Feb) of Mean Sea-Level Pressure (hPa) and total precipitation (mm). Mean Sea-Level Pressure is from NCEP and precipitation from SAFRAN (Aug. 1st 1981 to July 31st 2005).

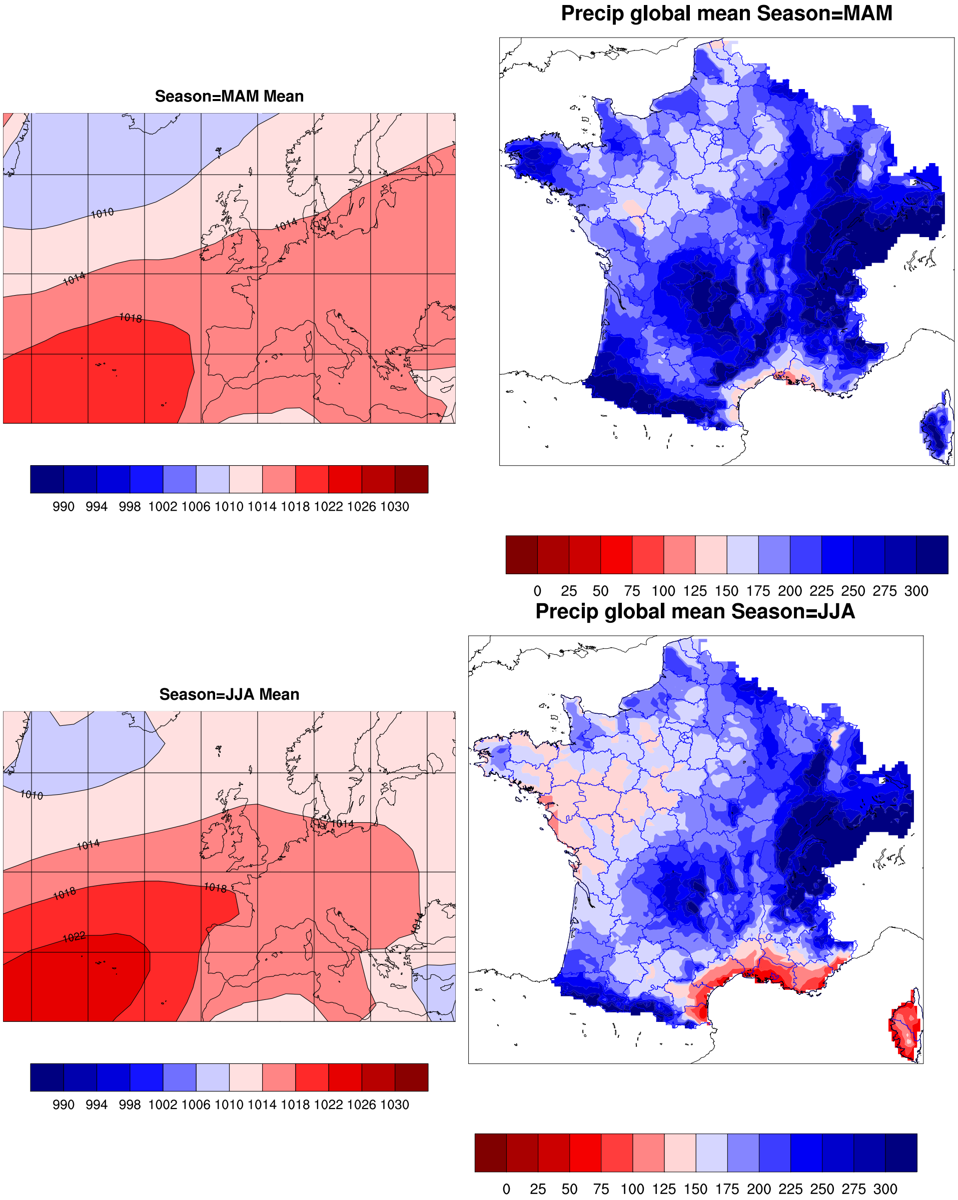

FIG. 4: Seasonal means over Spring (Mar-Apr-May) and Summer (Jun-Jul-Aug) of Mean Sea-Level Pressure (hPa) and total precipitation (mm). Mean Sea-Level Pressure is from NCEP and precipitation from SAFRAN (Aug. 1st 1981 to July 31st 2005).

From these seasonal climatologies, it is possible to derive weather types using a classification algorithm. The one used here is based on clustering analysis through the k-means automatic partitioning algorithm (Michelangeli et al., 1995). Classification is performed using the first ten principal components of an Empirical Orthogonal Functions (EOF) analysis, using both the MSLP and the square-root of precipitation. It must be noted that in this particular case, the ten principal components accounts for much of the observed variance (both for the MSLP and precipitation). For the final classification, only the weather-types corresponding to the centroids of MSLP are retained. Each day is then classified to the nearest weather-type. The major drawback for the k-means algorithm is that the number of clusters must be chosen a priori and that it is not a straightforward exercise to determine. One must be very careful when modifying the geographical region and the number of clusters.

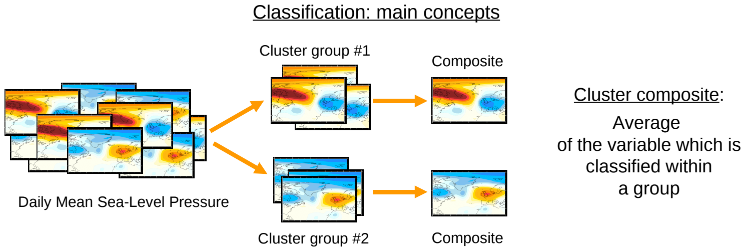

FIG. 5: Classification main concepts. The Daily Mean Sea-Level Pressure is the variable used in the classification for the Large-Scale Circulation as in Boé and Terray (2008a). *Figure courtesy of Julien Najac.*

To illustrate the dependence between MSLP and precipitation distribution anomalies, for each season separately, a composite of MSLP anomalies of the days belonging to each weather-type and the ratio between mean precipitation for the days within the weather-type and the global seasonal mean are shown (Figs 6; 7; 8 and 9).

These composites are very informative. It first shows that each weather type (MSLP anomaly) within a particular season (and with a particular LSC), there is a different response in terms of precipitation distribution over France. This is why in the current implementation of the methodology, there is the use of the four standard seasons. These are September-October-November (SON), December-January-February (DJF), March-April-May (MAM) and June-July-August (JJA).

The weather types shown here are the one derived from the current implementation of the methodology using the configuration shown later in the example case (see sect. 6.1). However, one must be careful not to directly associate these weather types with the known weather regimes (e.g. North Atlantic Oscillation (NAO-, NAO+), etc.), because they may not have specific distributions for a given local

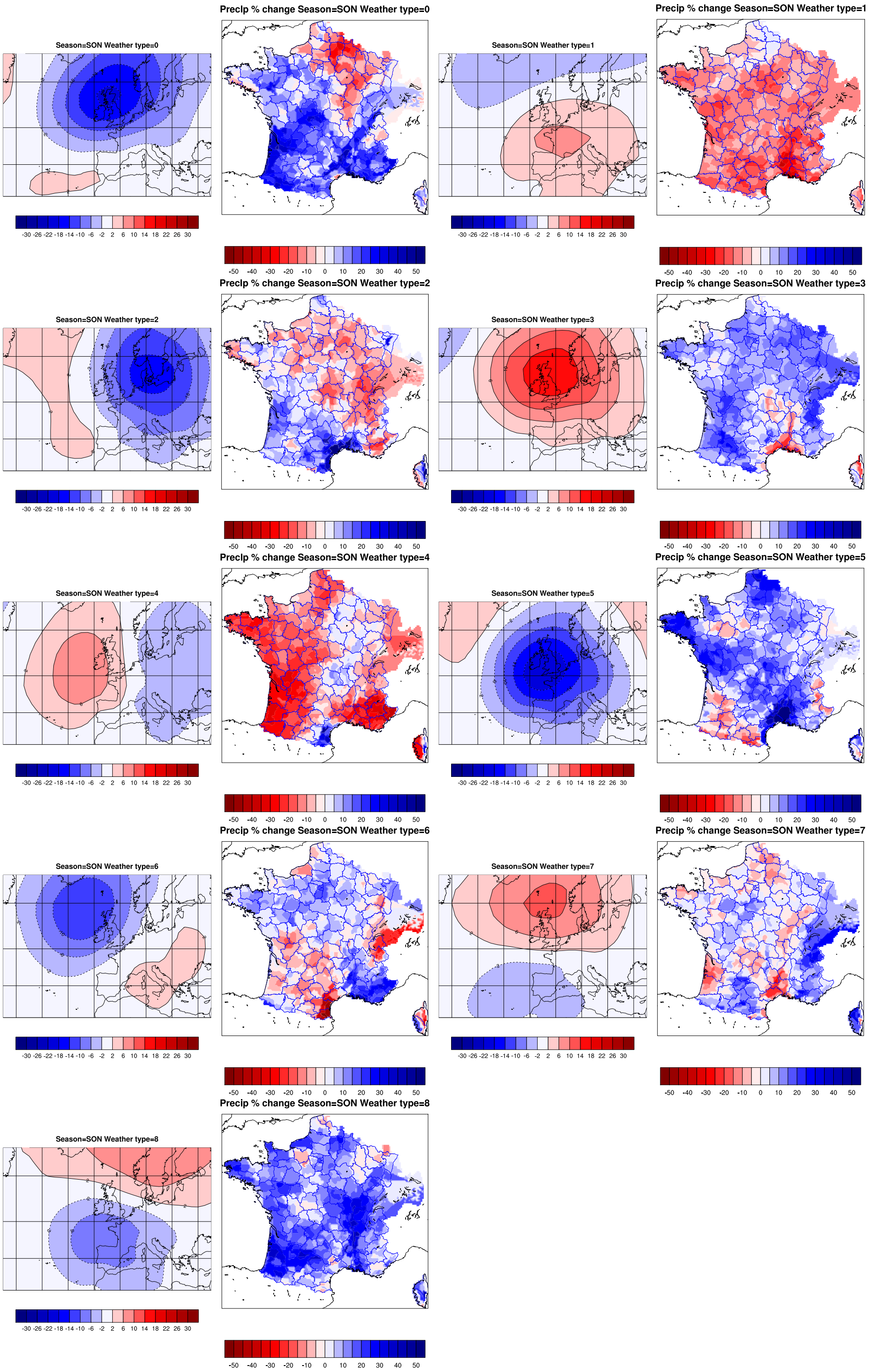
Season=SON Weather type=0
Precip % change Season=SON Weather type=0
Season=SON Weather type=1
Precip % change Season=SON Weather type=1
Season=SON Weather type=2
Precip % change Season=SON Weather type=2
Season=SON Weather type=3
Precip % change Season=SON Weather type=3
Season=SON Weather type=4
Precip % change Season=SON Weather type=4
Season=SON Weather type=5
Precip % change Season=SON Weather type=5
Season=SON Weather type=6
Precip % change Season=SON Weather type=6
Season=SON Weather type=7
Precip % change Season=SON Weather type=7
Season=SON Weather type=8
Precip % change Season=SON Weather type=8
-30 -26 -22 -18 -14 -10 -6 -2 2 6 10 14 18 22 26 30
-50 -40 -30 -20 -10 0 10 20 30 40 50

FIG. 6: Autumn (Sept-Oct-Nov) Weather types 0 to 8: Anomalies of Mean Sea-Level Pressure (hPa) and precipitation (%). Mean Sea-Level Pressure is from NCEP and precipitation from SAFRAN (Aug. 1st 1981 to July 31st 2005).

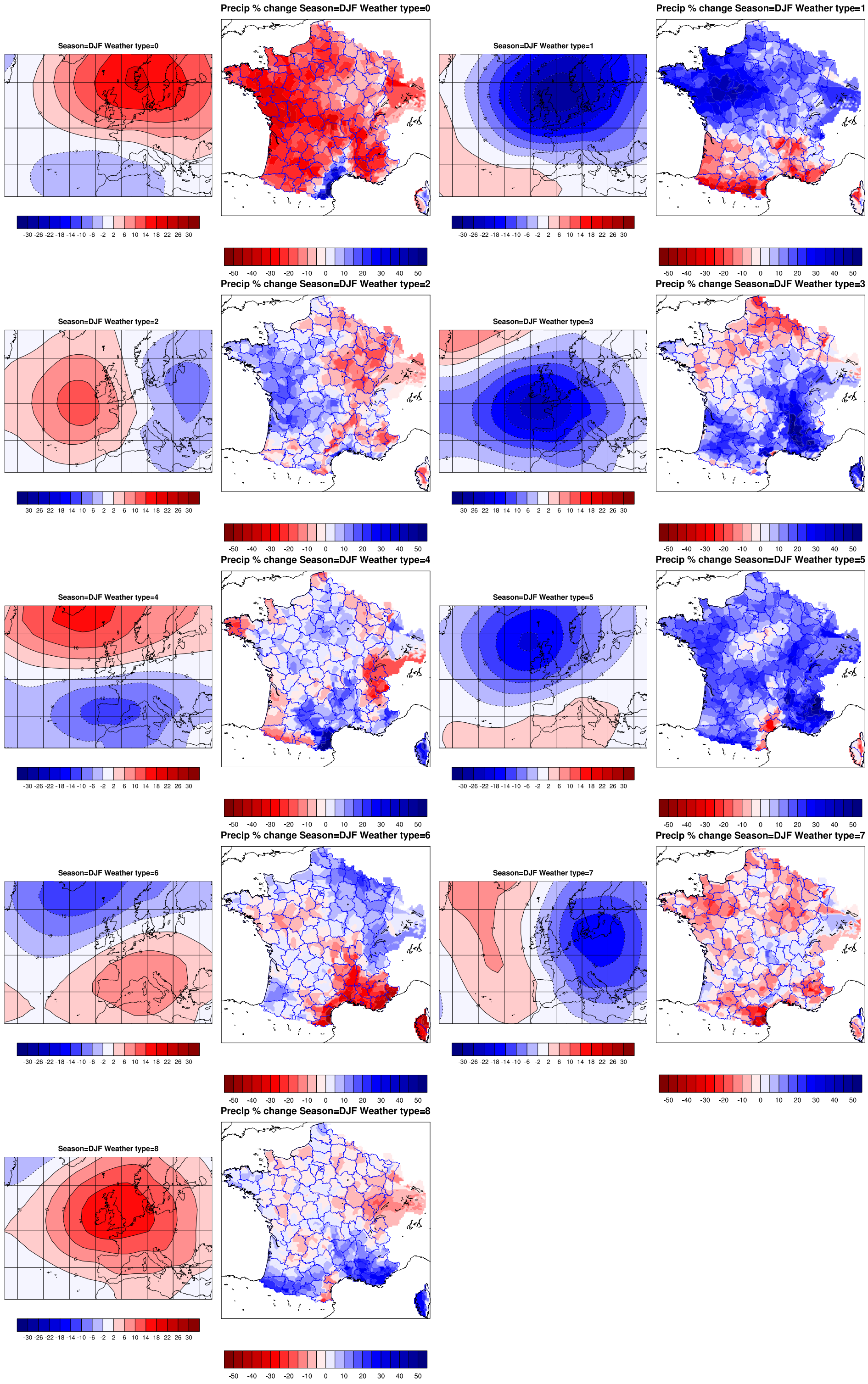
Season=DJF Weather type=0
Precip % change Season=DJF Weather type=0
Season=DJF Weather type=1
Precip % change Season=DJF Weather type=1
Season=DJF Weather type=2
Precip % change Season=DJF Weather type=2
Season=DJF Weather type=3
Precip % change Season=DJF Weather type=3
Season=DJF Weather type=4
Precip % change Season=DJF Weather type=4
Season=DJF Weather type=5
Precip % change Season=DJF Weather type=5
Season=DJF Weather type=6
Precip % change Season=DJF Weather type=6
Season=DJF Weather type=7
Precip % change Season=DJF Weather type=7
Season=DJF Weather type=8
Precip % change Season=DJF Weather type=8
-30 -26 -22 -18 -14 -10 -6 -2 2 6 10 14 18 22 26 30
-50 -40 -30 -20 -10 0 10 20 30 40 50

FIG. 7: Winter (Dec-Jan-Feb) Weather types 0 to 8: Anomalies of Mean Sea-Level Pressure (hPa) and precipitation (%). Mean Sea-Level Pressure is from NCEP and precipitation from SAFRAN (Aug. 1st 1981 to July 31st 2005).

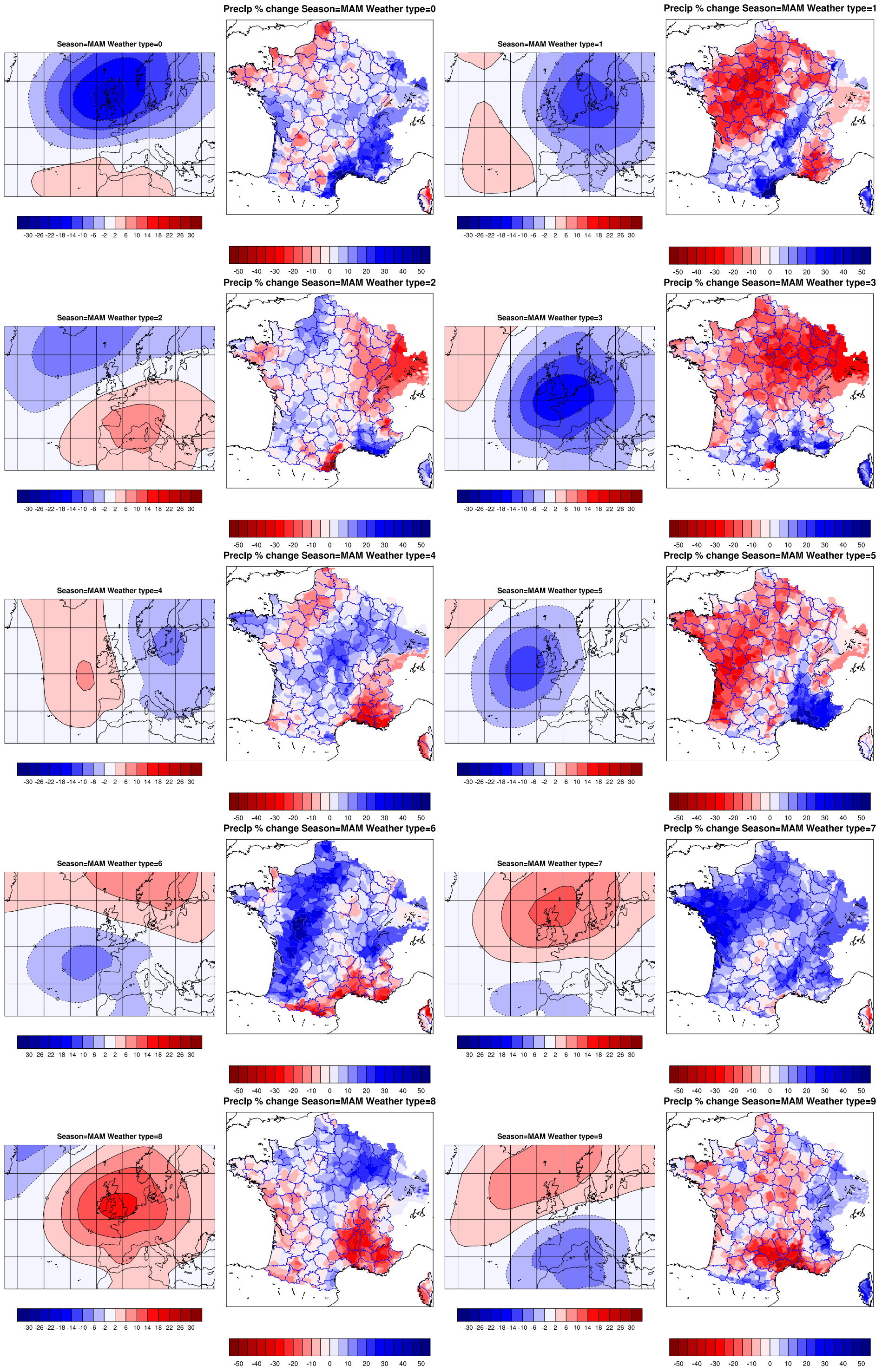
Season=MAM Weather type=0
Precip % change Season=MAM Weather type=0
Season=MAM Weather type=1
Precip % change Season=MAM Weather type=1
Season=MAM Weather type=2
Precip % change Season=MAM Weather type=2
Season=MAM Weather type=3
Precip % change Season=MAM Weather type=3
Season=MAM Weather type=4
Precip % change Season=MAM Weather type=4
Season=MAM Weather type=5
Precip % change Season=MAM Weather type=5
Season=MAM Weather type=6
Precip % change Season=MAM Weather type=6
Season=MAM Weather type=7
Precip % change Season=MAM Weather type=7
Season=MAM Weather type=8
Precip % change Season=MAM Weather type=8
Season=MAM Weather type=9
Precip % change Season=MAM Weather type=9
-30 -26 -22 -18 -14 -10 -6 -2 2 6 10 14 18 22 26 30
-50 -40 -30 -20 -10 0 10 20 30 40 50

FIG. 8: Spring (Mar-Apr-May) Weather types 0 to 8: Anomalies of Mean Sea-Level Pressure (hPa) and precipitation (%). Mean Sea-Level Pressure is from NCEP and precipitation from SAFRAN (Aug. 1st 1981 to July 31st 2005).

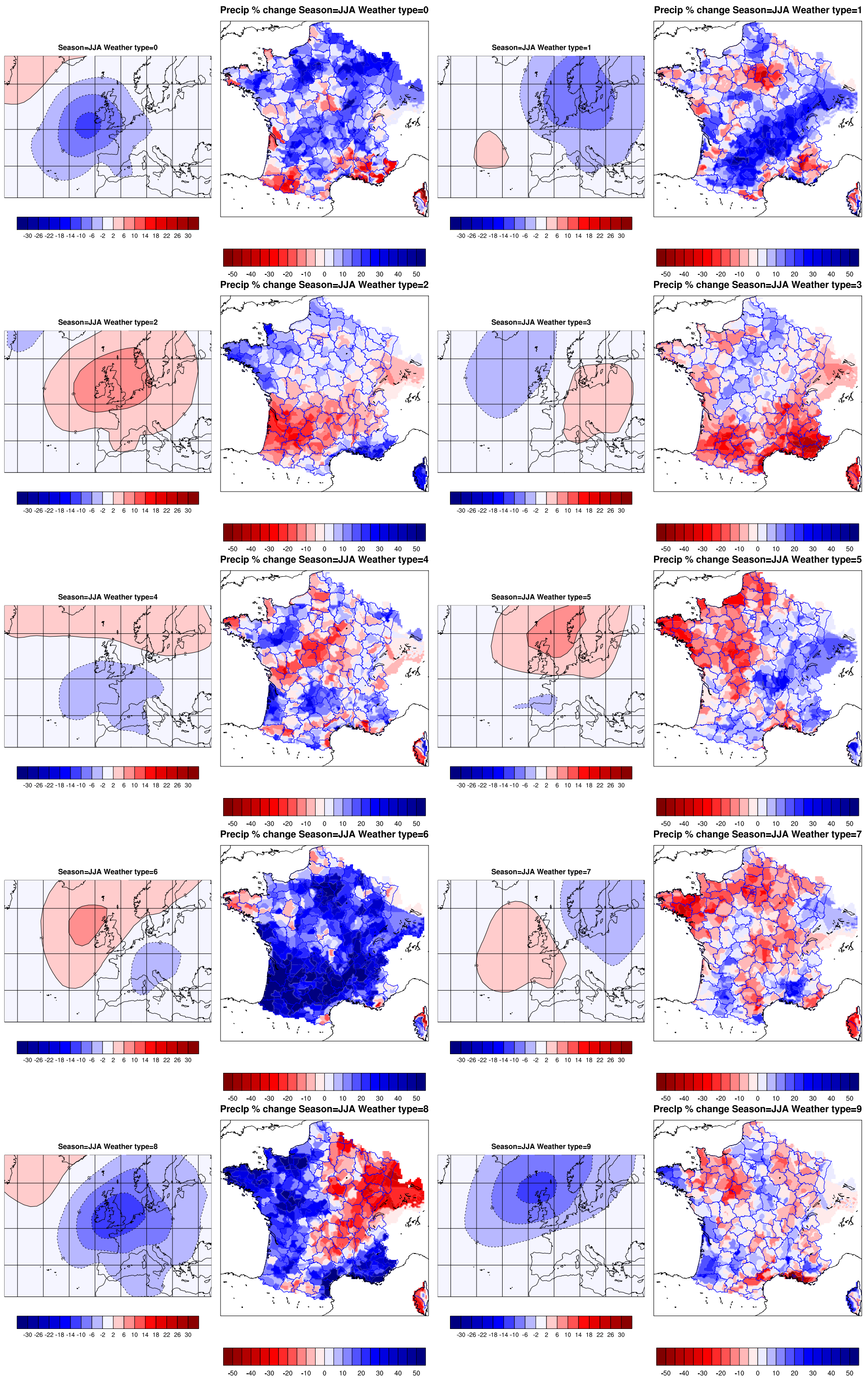
Season=JJA Weather type=0
Precip % change Season=JJA Weather type=0
Season=JJA Weather type=1
Precip % change Season=JJA Weather type=1
Season=JJA Weather type=2
Precip % change Season=JJA Weather type=2
Season=JJA Weather type=3
Precip % change Season=JJA Weather type=3
Season=JJA Weather type=4
Precip % change Season=JJA Weather type=4
Season=JJA Weather type=5
Precip % change Season=JJA Weather type=5
Season=JJA Weather type=6
Precip % change Season=JJA Weather type=6
Season=JJA Weather type=7
Precip % change Season=JJA Weather type=7
Season=JJA Weather type=8
Precip % change Season=JJA Weather type=8
Season=JJA Weather type=9
Precip % change Season=JJA Weather type=9
-30 -26 -22 -18 -14 -10 -6 -2 2 6 10 14 18 22 26 30
-50 -40 -30 -20 -10 0 10 20 30 40 50

FIG. 9: Summer (Jun-Jul-Aug) Weather types 0 to 8: Anomalies of Mean Sea-Level Pressure (hPa) and precipitation (%). Mean Sea-Level Pressure is from NCEP and precipitation from SAFRAN (Aug. 1st 1981 to July 31st 2005).

scale variable even if they exhibit some similarities.

1.2 Specific implementation

In the current software implementation, the large-scale predictors are generalized, which means that any large-scale field could be used. However, they are necessary linked by a regression analysis to the total precipitation accumulation, which is a fixed predictant in the current implementation. In Boe08's methodology, the main predictor is the Mean Sea-Level pressure with Temperature at 2 m used as a secondary large-scale predictor for specific seasons. The secondary large-scale predictor is also used in the final analog day selection, while the local-scale predictant is the total precipitation accumulation. A current limitation of the implementation is that the learning process can only be performed using a gridded observation database. This limitation could be eliminated in future versions.

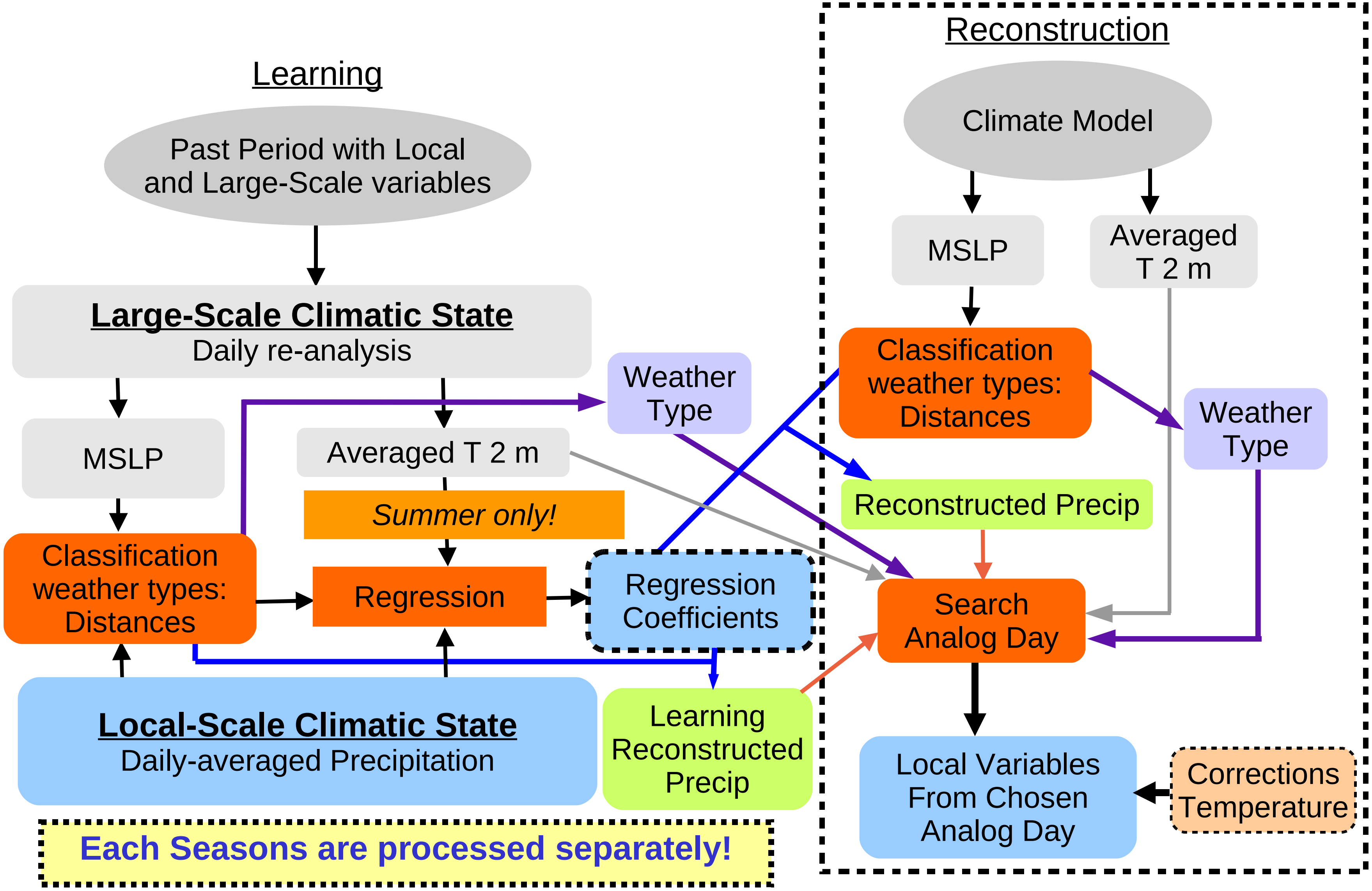
Learning
Past Period with Local and Large-Scale variables
Large-Scale Climatic State
Daily re-analysis
MSLP
Averaged T 2 m
Summer only!
Classification weather types: Distances
Regression
Regression Coefficients
Weather Type
Local-Scale Climatic State
Daily-averaged Precipitation
Learning Reconstructed Precip
Each Seasons are processed separately!
Reconstruction
Climate Model
MSLP
Averaged T 2 m
Classification weather types: Distances
Weather Type
Reconstructed Precip
Search Analog Day
Local Variables From Chosen Analog Day
Corrections Temperature

FIG. 10: Complex Schema describing the specific implementation of Boé and Terray (2008a) statistical downscaling methodology.

The general methodology is presented in figure 10. This relatively complex schema gives a nice overview of the whole methodology main structure, as well as the relationship between the learning and downscaling processes. Let's describe this schema in more details in the following paragraphs. It must be noted that each season is processed separately.

First, it is important to mention that it is necessary to have daily data time series covering a long time period and being ideally homogeneous. The observed dataset should span a large sample of climate conditions, including future modifications of LSC predictors. These time series must provide both the atmospheric LSC variables (predictors) and the climatic local variables (predictants) to be able to generate the statistical transfer function (the learning process).

The **learning phase** of the methodology is performed on a past period (long enough, typically 25 or more years) in which one have both large-scale and local-scale data. In this case here, the large-scale data fields used are the Mean Sea-Level Pressure (MSLP) and the 2-meter Air Temperature. These fields are from the NCEP daily re-analysis over the LSC region (Fig. 2). The MSLP (anomaly and with climatology removed) is used in the classification algorithm along with the observed local-scale daily-averaged precipitation to determine the weather types, their respective cluster centres, and assign each day in the learning period to the closest weather type (Euclidian distance). This information is then used to calculate the distances to all the weather types for each day in the learning period. In parallel, the large-scale daily temperature is spatially-averaged over the whole region (optionally with a mask to average only over land). The distances to the weather types are then used in a regression equation (along with the averaged temperature, for summer season only) with the local-scale precipitation to calculate the regression coefficients. These statistically relate the distances to all the weather types to a given spatial local-scale precipitation structure. These regression coefficients will be used in the downscaling part of the algorithm. They are also used in the last step of the learning process to recalculate the local-scale precipitation over the regression points using the newly determined coefficients. This will be used when choosing the analog day, as a proxy to the LSC anomaly patterns.

The **downscaling phase** involves the use of the output of large-scale fields which we want to downscale (generally data from a climate model simulation). From this output, the large-scale fields of MSLP anomaly and 2-meter Air Temperature are selected over the same region that was used in the learning process. The temperature is spatially-averaged (as in the learning process), and the MSLP anomaly is used to calculate, for each downscaled day, the distances to all the weather types (which were determined in the learning process), and the associated weather type. With the previously calculated regression coefficients, these distances are used to calculate a reconstructed precipitation over the regression points. This reconstructed precipitation is then used to search the analog day (as a proxy to the LSC) by comparing this precipitation to each day of the learning period. Along with this comparison, the algorithm also compares the spatially-averaged temperature. These two criterias are used to select the analog day, by searching in the learning period only those days having the same weather type. Depending on the configuration of the algorithm, the method to choose the analog day can differ, as there are many different ways to choose the "closest" day.

When the day is chosen, all the data from the observation database are read and written for the given downscaled day. However, when there is an absolute temperature difference greater than 2 degrees C between the spatially-averaged temperature of the climate model output compared to the analog day re-analysis temperature, the analog day data is corrected by the difference of temperature. Eventually if available in the observation database and selected for output, infra-red radiation, relative humidity, solid/liquid precipitation partition and potential evapotranspiration are then corrected as well.

It must be noted that in the classification algorithm, the MSLP anomaly (with climatology removed) is decomposed using an Empirical Orthogonal Functions (EOF) analysis in the learning phase, while in the downscaling phase, the MSLP anomaly field is projected onto the pre-calculated EOFs. For local-scale precipitation, which is also used in the determination of the weather types in the learning phase, it is also decomposed using an EOF analysis. It must also be stressed that the algorithm also highly depends on proper normalisation all along the two phases. Finally, the field of precipitation used throughout the algorithm is the square-root of precipitation. For more in-depth discussion of Boe08's methodology, the reader is referred to Boé (2007); Boé and Terray (2008a,b); Boé et al. (2006) papers.

1.3 Validation

The methodology has been validated extensively by Julien Boé (Boé, 2007; Boé and Terray, 2008a,b; Boé et al., 2006). The main validations are shown in this section.

The three main hypotheses of statistical downscaling methodologies are:

1. Selected LSC climate predictors are appropriate variables for the problematic studied (regional/local climate), and are simulated realistically by climate numerical models;
2. Stationarity: the statistical relationship is still valid for the perturbed climate (by anthropic and/or natural forcings). However, this hypothesis cannot be verified or invalidated formally.

- Dynamical downscaling has a similar hypothesis
  - Physical Parameterisations
  - Bias correction (quantile-quantile)

3. Predictors react to climate change signal

First, the weather type frequencies of occurrence were compared over the 1950-1999 period, between the NCEP re-analysis and many climate models. The results shown here are from a simulation of the Météo-France ARPEGE climate model (Salas y Mélia et al., 2005). The four standard seasons are shown separately (see Fig. 11). It follows that there is a very good agreement between this ARPEGE simulation and the NCEP re-analysis over the 1950-1999 period, regarding the frequency of occurrence of weather types.

Another important aspect, as explained in details in Boe08, is that if a simple model is built using the weather types occurrence frequency and covariances as predictors, the methodology must be able to capture an important part of the inter-annual variability of precipitation and the spatial pattern of precipitation trends. This validation test as been done over the second half of the twentieth century (Fig. 12), and it gives satisfactory results.

When downscaling some AR4 IPCC climate models over the period 1981-2000 and comparing the resulting seasonal mean of precipitation with the observed local-scale precipitation, the mean absolute error is in the range of 3 to 14% (Fig. 13).

The stationarity hypothesis can be validated somewhat informally, using a perfect model experiment. The idea is to use the model precipitation and mean sea-level pressure as re-analysis and observation in the learning process, for the period 1950-1999, then to downscale an independent model simulation of the same model. Thus, it permits to evaluate if the statistical relationships derived in the past (by the methodology) can be applied in a modified climate as simulated by a climate model (Fig. 14).

The ability of the methodology to reproduce the past observed precipitation using downscaled NCEP re-analysis LSC is very important. It shows the maximum performance of the algorithm to reproduce observed precipitation structures for a given period (Fig. 15). Because the algorithm configuration has been adjusted to perform quite well over the whole France (especially the LSC domain), the results performance can vary from one region to another. This may be the cause of the lesser ability of the algorithm over the Meditterranean coast region.

In conclusion, the validation of the algorithm in this configuration is quite satisfactory. All the weather types are discriminant enough, the relative errors are limited, the past precipitation tendencies are well reproduced, the stationary hypothesis is

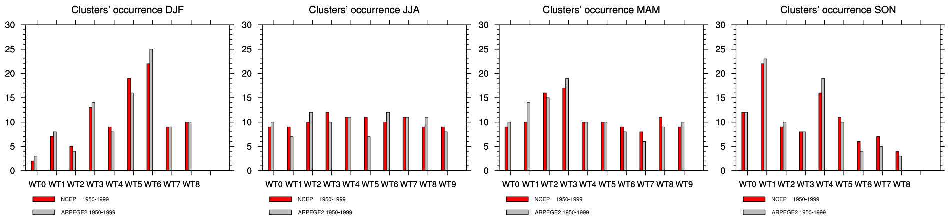

FIG. 11: Comparison of the frequency of occurrence (%) of each weather type (WT) for the period 1950-1999. NCEP re-analysis is in red, while Météo-France ARPEGE climate model is in gray. The four standard seasons are shown separately. These are September-October-November (SON), December-January-February (DJF), March-April-May (MAM) and June-July-August (JJA).

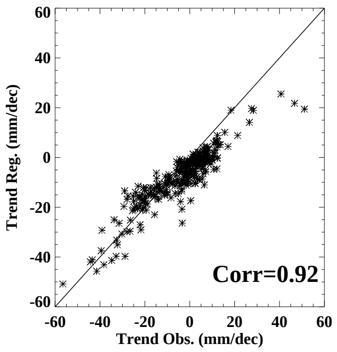

FIG. 12: Observed vs reconstructed by regression linear trends (mm per decade) in precipitation for the 1951-2000 period. The black line corresponds to the equation y=x. WT occurrence frequency and covariances as predictors. *From Boé and Terray (2008a) Fig. 14a.*

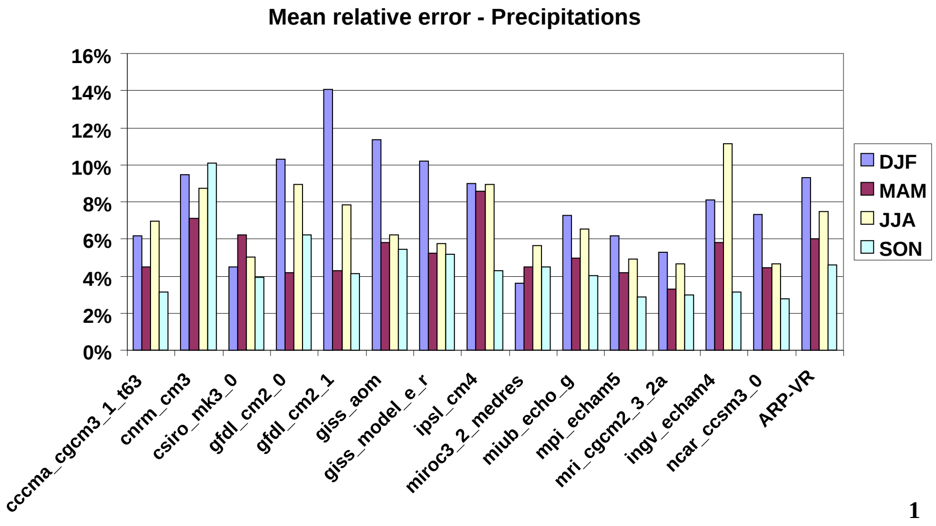

FIG. 13: Observed vs reconstructed precipitation seasonal mean error when using AR4 IPCC model large-scale predictors. Period 1981-2000. *Figure courtesy of Julien Boé.*

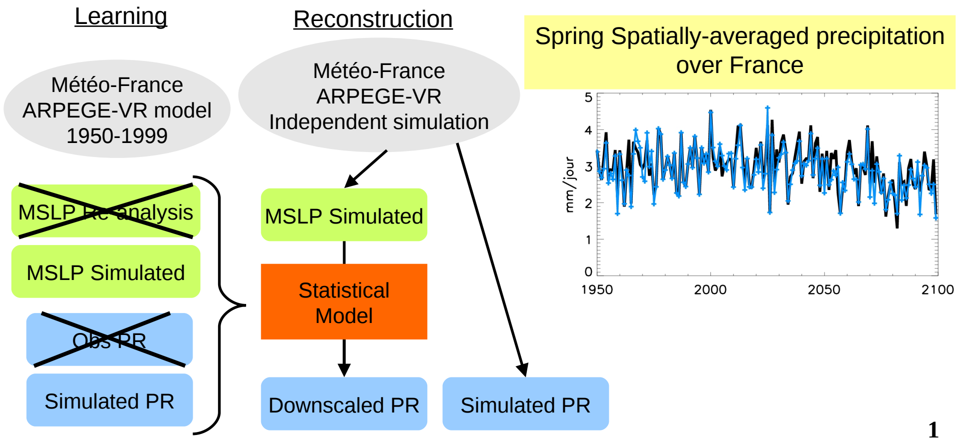

FIG. 14: Perfect model experiment. The precipitation axis is in mm/day. Black curve is the reconstructed precipitation control run, and the blue curve is the reconstructed precipitation independent model run. *Figure courtesy of Julien Boé.*

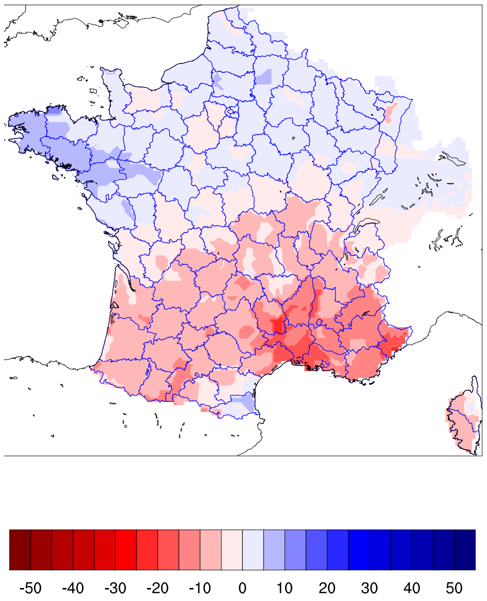

FIG. 15: Difference downscaled NCEP vs SAFRAN of mean annual total precipitation (%) over France over the period 1971-1998.

informally verified, and the ARPEGE climate model simulates correctly the occurrence of the weather types over each season.

The next section will discuss precisely the structure of the algorithm and the different steps of the methodology implementation as well as more technical details.

2 General structure

The methodology implementation has been structured and generalized somewhat to enable different configurations of the methodology. This section will both present the general structure and the steps in the methodology implementation.

The structure can be summarized as follows:

1. Load algorithm configuration from file
2. Read regression points positions
3. Learning process using reanalysis large-scale atmospheric fields and local-scale observations database
   - Can be computed or just read off disk if pre-computed already.

4. Downscaling process
    - Can be computed or just read off disk if pre-computed already.
5. Read analog day data and write data to downscaled day, for control period and downscaled period

The main detailed structure is quite lengthy:

1. Load algorithm configuration from file
2. Read regression points positions
3. Learning process using reanalysis large-scale atmospheric fields and local-scale observations database.
    - Read re-analysis pre-computed Empirical Orthogonal Functions (EOF) and singular values
    - Read observations pre-computed EOF and singular values
    - Select common time period between the re-analysis and the observation data periods
    - Compute normalisation factor of EOF of large-scale circulation field for the whole period
        - Renormalize EOF of large-scale circulation field for the whole period using the first EOF norm and the Singular Value
        - Recompute normalization factor using normalized field
    - Read observed precipitation (liquid and solid) and compute total precipitation
        - Mask part of the region (if needed) for regression analysis
    - Perform spatial mean of observed precipitation around regression points
    - Normalize precipitation
    - Read re-analysis and observation data for secondary large-scale field
        - Select common time period
        - Extract subdomain
    - Perform spatial mean of secondary large-scale fields
    - For each season separately
        - Select season months in the whole time period and create sub-period fields
        - Normalize secondary large-scale fields for re-analysis learning data
            - Compute mean and variance over time
            - Normalize using mean and variance
    - Merge observation and reanalysis EOF principal components for clustering algorithm
        - Normalize using corresponding first Singular Value
    - Compute best clusters using Michelangeli et al. (1995) classification algorithm

- Classify each day in the closest clusters
- Compute normalized distances to clusters
- Compute regressions coefficients and constant on regression points
  - Regression between:
    (a) Distance to clusters and, optionally for special seasons, using a secondary large-scale field
    (b) Total precipitation
  - Recompute total precipitation using calculated regression coefficients and constant
- Write learning data to files

4. Downscaling process
   - Read large-scale fields for control period and downscaled period
   - Compute and remove climatologies from selected large-scale fields using climatology of control period
   - Read pre-computed EOF and singular values
   - Project large-scale circulation field onto pre-computed EOF and singular values
   - Prepare data for downscaling (normalization, distance to clusters, secondary large-scale fields)
     - Process Control period
       - ∗ Normalize the large-scale circulation field principal component by the standard deviation of the first one
       - ∗ Select common time period between the learning period and the model period (control period)
       - ∗ Compute the norm and the variance of the first EOF of the control period as references
       - ∗ Normalize the large-scale circulation EOF-projected field given the reference norm and variance
       - ∗ For each season separately
         - · Select common time period between the learning period and the model period (control period)
         - · Compute seasonal mean and variance of distances to clusters
         - · Compute seasonal mean and variance of spatially-averaged secondary large-scale fields
         - · Normalize the spatially-averaged secondary large-scale fields
         - · Apply the regression coefficients to calculate precipitation using:
         (a) Cluster distances
         (b) Normalized spatial mean of the corresponding secondary large-scale field

    - Optionally downscale the control period, downscale the downscaled period
      - Compute seasonal mean and variance of spatially-averaged secondary large-scale fields
      - Normalisation of the large-scale circulation field principal component by the standard deviation of the control period
      - For each season separately

      (a) Compute distances to clusters using normalization and against the control reference period
      (b) Classify each day in the best clusters
      (c) Normalize the spatially-averaged secondary large-scale fields
      (d) Apply the regression coefficients to calculate precipitation using:
      (e) Cluster distances and normalized spatial mean of the corresponding secondary large-scale field

- Resampling: find analog days for control period (optional) and downscaled period
  - For each season separately
    - Select correct months for the current season of the downscaled and learning period
    - Process each downscaled day separately:
    - Search analog days in learning period, only those having the same associated cluster
      - Compute squared precipitation index difference
      - Compute square root of squared precipitation index (precipitation metric)
      - Compute standard deviation and mean of precipitation metric
      - Apply normalization to precipitation metric
      - Optionally, use the secondary large-scale fields in the first selection of days

      (a) Compute secondary large-scale field index difference (secondary metric)
      (b) Compute square root of squared secondary large-scale field index difference
      (c) Compute standard deviation and mean of secondary metric
      (d) Apply normalization to secondary metric
      (e) Sum both cluster and secondary metrics

      - Select specified number of days within specified range
      - Depending on configuration options

      (a) Select a random day for this second and final selection

(b) Choose the day having the smallest metric for the best match:

(c) Either use the secondary large-scale field index for having the smalles metric

(d) or use the main large-scale field (precipitation) as the metric for the final selection

∗ For the selected analog day, compute the secondary large-scale field difference (temperature change) between model downscaled day and re-analysis analog day

5. Read analog day data and write data to downscaled day, for control period (optional) and downscaled period
    - Process each observation variable
        - Retrieve temperature change. If it is greater than specified value:
            - ∗ Adjust temperature (or min and max temperatures)
            - ∗ Adjust infra-red radiation if available
            - ∗ Adjust liquid and solid precipitation if available
            - ∗ Adjust relative humidity if available
        - Write each observation variable to downscaled day

The main algorithm will be described in more details in the following subsections. The configuration file is a well-structured XML file. Its file syntax and structure and the associated configuration parameters will be discussed in section 4.

2.1 Regression

The regression points must be defined in the region of interest where observation data is available. In Boe08, 220 regression points were chosen on Meteo-France ARPEGE-CLIMAT (Salas y Mélia et al., 2005) grid covering the Metropolitan France, and observation data points were aggregated onto these points (see Fig. 16). The regression points does not have to be on the climate model grid, but they must at least cover all the area of the predictant observation variable. In dsclim, these points are specified using a NetCDF (Russell et al., 2006) input file. The NetCDF regression points file must follow the convention shown in Listing 1. This convention specifies that the latitude and longitude vectors are one-dimensional.

2.2 Learning process

The learning process should not be performed each time downscaling is done, especially noting the fact that each learning process will yield to slightly different

Listing 1: NetCDF file convention for regression points

```
netcdf reg_pts_france {

  dimensions:
    pts = UNLIMITED ; // (220 currently)

  variables:
    float lat(pts) ;
      lat:units = "degrees_north" ;
      lat:long_name = "latitude" ;
      lat:standard_name = "latitude" ;
    float lon(pts) ;
      lon:units = "degrees_east" ;
      lon:long_name = "longitude" ;
      lon:standard_name = "longitude" ;
}
```

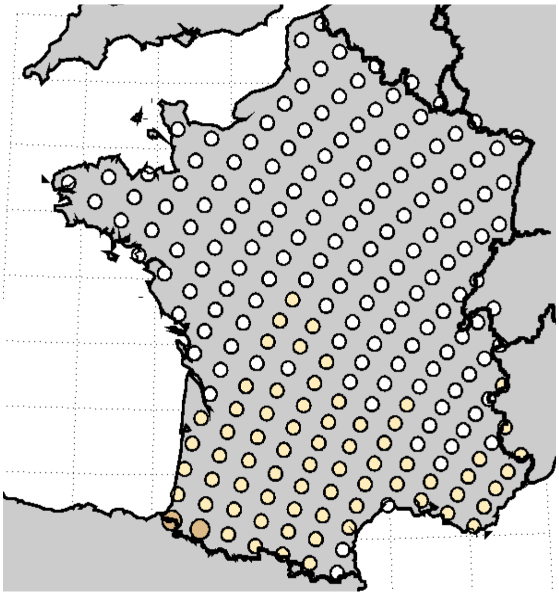

FIG. 16: The 220 regression points used in Boe08.

Listing 2: NetCDF file convention for EOFs

```
netcdf psl_1d_19480101_20060331_NCP_aseason_EOF {
dimensions:
        time = UNLIMITED ; // (21275 currently)
        lon = 144 ;
        lat = 73 ;
        eof = 10 ;
variables:
        int time(time) ;
                time:units = "days since 1948-01-01T12:00:00" ;
                time:calendar = "gregorian" ;
                time:long_name = "time in days since 1948-01-01T12:00:00" ;
        float lon(lon) ;
        float lat(lat) ;
        float psl_eof(eof, lat, lon) ;
                psl_eof:long_name = "Empirical Orthogonal Functions" ;
                psl_eof:short_name = "EOFs" ;
                psl_eof:missing_value = 1.e+20 ;
        float psl_pc(time, eof) ;
                psl_pc:long_name = "Principal Components" ;
                psl_pc:short_name = "PCs" ;
        double psl_sing(eof) ;
                psl_sing:long_name = "Singular Values" ;
                psl_sing:short_name = "SingVals" ;
}
```

results because some randomness is involved in the classification methodology. Consequently, the learning process should only be done when modifications are needed in the predictors or predictants, in the learning time period, or in the spatial region that is considered by the algorithm configuration. Learning data can be saved into NetCDF files to be read subsequently, when no change is needed in the above mentioned parameters.

In dsclim, it is assumed that re-analysis and predictant EOFs are already precomputed, and that they follow the conventions of the STATPACK (Pascal Terray) EOF norms. The pre-computed EOFs must follow the NetCDF convention as shown in Listings 2. The important point is that there must be variables for Empirical Orthogonal Functions (3D), Principal Components (2D) and Singular Values (1D), along with time, longitude and latitude variables, as well as time (unlimited), longitude, latitude and eof dimensions. Also, it must be stored as one file for the whole period. The actual names of the variables is configurable in the configuration file, as it will be shown in section 4.

One must be **very careful** that the time variable be correct: it is very important!

The learning period considered in the algorithm is the whole overlap period of both the re-analysis and observations pre-computed EOFs. The spatial domain must be the same as the one that will be used in the downscaling process later on. The spatial domain for the large-scale circulation field can be different from the one used for the secondary large-scale field (the temperature in Boe08's methodology). The latter spatial domain can have a mask applied to it to only consider relevant gridpoints, such as only land points by example. The domain used in Boe08 is shown in Fig. 2.

Listing 3: NetCDF file convention for re-analysis data

```
netcdf tas_1d_19480101_20070331_NCP {
dimensions:
        time = UNLIMITED ; // (21640 currently)
        lat = 94 ;
        lon = 192 ;
variables:
        float tas(time, lat, lon) ;
                tas:missing_value = 1.e+20f ;
                tas:units = "degK" ;
                tas:long_name = "mean Daily Air temperature at 2 m" ;
                tas:_FillValue = 1.e+20f ;
        double time(time) ;
                time:units = "hours since 1-1-1 00:00:0.0" ;
                time:long_name = "Time" ;
                time:delta_t = "0000-00-01 00:00:00" ;
                time:avg_period = "0000-00-01 00:00:00" ;
        float lat(lat) ;
                lat:units = "degrees_north" ;
                lat:long_name = "Latitude" ;
        float lon(lon) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
}
```

The re-analysis secondary large-scale field used as input in the learning process must also follow a simple NetCDF convention where, as shown in Listing 3, the only requirements being that the variable has dimensions (time, lat, lon), that these dimensions also exists as variables, and that it is on a regular Latitude-Longitude grid. The other requirement is that re-analysis data must be stored in files covering the whole time period. Again, one must be **very careful** that the time variable be correct, which is not always the case!

The observation database fields used as input in the learning process must follow the NetCDF CF-1.0 convention (Program for Climate Model Diagnosis and Intercomparison, 2008), except for the variable names which are configurable in the configuration file. The observation variables must be 3D (time, y, x), and the latitude and longitude variables must be either 1D (Latitude-Longitude projection) or 2D (Latitude-Longitude or Lambert-Conformal projections) on a regular grid. In Boe08's implementation, the SAFRAN (Quintana-Seguí et al., 2008) analysis has been used (see Listing 4). The regular grid projection of SAFRAN is Lambert Conformal.

The regression in the learning process can also have a mask region applied to it. In Boe08's methodology, the algorithm is applied to Metropolitan France, but Corsica is excluded from the learning process when computing regressions for precipitation. The regression region mask is specified in the configuration file as a bounding box. In future versions a mask specifying individual points will be implemented.

The classification is performed using the Michelangeli et al. (1995) classification algorithm, using euclidian distance. The classification methodology involves starting with a random selection of cluster centers, which means that there is some randomness of the algorithm. It follows that if the learning process is performed

Listing 4: Example NetCDF file convention for observation data

```
netcdf ForcPRCP.DAT_france_6465_daily {
dimensions:
        time = UNLIMITED ; // (365 currently)
        x = 143 ;
        y = 134 ;
variables:
        int Lambert_Conformal ;
                Lambert_Conformal:grid_mapping_name = "lambert_conformal_conic" ;
                Lambert_Conformal:standard_parallel = 45.89892f, 47.69601f ;
                Lambert_Conformal:longitude_of_central_meridian = 2.337229f ;
                Lambert_Conformal:latitude_of_projection_origin = 46.8f ;
                Lambert_Conformal:false_easting = 600000.f ;
                Lambert_Conformal:false_northing = 2200000.f ;
        int x(x) ;
                x:units = "m" ;
                x:long_name = "x coordinate of projection" ;
                x:standard_name = "projection_x_coordinate" ;
        int y(y) ;
                y:units = "m" ;
                y:long_name = "y coordinate of projection" ;
                y:standard_name = "projection_y_coordinate" ;
        double lon(y, x) ;
                lon:units = "degrees_east" ;
                lon:long_name = "longitude coordinate" ;
                lon:standard_name = "longitude" ;
        double lat(y, x) ;
                lat:units = "degrees_north" ;
                lat:long_name = "latitude coordinate" ;
                lat:standard_name = "latitude" ;
        short Altitude(y, x) ;
                Altitude:units = "meters" ;
                Altitude:long_name = "Altitude" ;
                Altitude:standard_name = "Altitude" ;
                Altitude:grid_mapping = "Lambert_Conformal" ;
                Altitude:coordinates = "lon lat" ;
                Altitude:missing_value = -99s ;
        int time(time) ;
                time:units = "hours since 1900-01-01T00:00:00Z" ;
                time:long_name = "time in hours since 1900-01-01T00:00:00Z" ;
        float PRCP(time, y, x) ;
                PRCP:long_name = "Liquid Precipitation" ;
                PRCP:units = "kg m-2 s-1" ;
                PRCP:grid_mapping = "Lambert_Conformal" ;
                PRCP:coordinates = "lon lat" ;
                PRCP:missing_value = -9999.f ;

// global attributes:
                :Conventions = "CF-1.0" ;
                :Metadata_Conventions = "Unidata Dataset Discovery v1.0" ;
                :title = "Meteo-France SAFRAN data" ;
                :title_french = "Donnees SAFRAN Meteo-France" ;
                :summary_french = "Donnees SAFRAN Meteo-France" ;
                :summary = "Meteo-France SAFRAN data" ;
                :keywords = "climat,SAFRAN,Meteo-France,Cerfacs" ;
                :processor = "IDL scripts" ;
                :description = "Meteo-France SAFRAN data" ;
                :cdm_datatype = "Grid" ;
                :institution = "Cerfacs" ;
                :creator_email = "globc@cerfacs.fr" ;
                :creator_url = "http://www.cerfacs.fr/globc/" ;
                :creator_name = "Global Change Team" ;
                :time_coverage_start = "1964-08-01T00:00:00Z" ;
                :time_coverage_end = "1965-07-31T23:59:59Z" ;
                :timestep = "daily" ;
                :contact_email = "christian.page@cerfacs.fr" ;
                :contact_name = "Christian PAGE" ;
                :other_contact_email = "laurent.terray@cerfacs.fr" ;
                :other_contact_name = "Laurent TERRAY" ;
                :geospatial_lat_max = 51.1215157941104 ;
                :geospatial_lat_min = 41.3181733938225 ;
                :geospatial_lon_max = 10.7928123474121 ;
                :geospatial_lon_min = -5.32915830612183 ;
}
```

each time, it is sure that results will differ slightly and are not reproducible, unless the learning data is saved on this for subsequent use. When the learning process is not performed, learning data is read from previously written NetCDF files by dsclim which ensures results reproducibility.

2.3 Downscaling process

The downscaling process itself is also optional, which means that if downscaling was previously performed and the analog dates along with the temperature change were saved by dsclim, the algorithm can be used to not perform the downscaling calculations but rather only output the downscaled data using the observation database.

The downscaling is performed in two main steps (processing each season separately). In order to use the learning data, it is necessary to normalize properly the climate model data. To do this, model data must be available for the same spatial domain and time period as the learning period (the control period), which is limited by the common time periods and spatial domains of both re-analysis and observation data. Consequently, the first steps after reading pre-computed EOFs is to compute the mean and standard deviation of both the distances to the clusters and the spatial average of the secondary large-scale field. The second step is to downscale both the control period and another period. Currently, only one other period can be downscaled, but the control period can be optionally downscaled.

There are several input files necessary to perform the downscaling process. The large-scale fields are read from NetCDF files, in the same convention as shown in Listing 3. The removal of the climatologies from the circulation large-scale field is done using a hanning filter algorithm (with wrapping edges in time). The climatologies are calculated using the control period. The pre-computed EOF and Singular Values are read as in the learning process (see Listing 2). A last requirement is that datafiles must be stored in files covering the whole time period.

The downscaling itself is performed by computing distances to clusters normalized against the control period. These clusters distances are then used to classify each day in the best clusters which were computed in the learning process. Then, the spatially-averaged secondary large-scale fields are normalized, and the total precipitation is calculated using the regression coefficients which are using both the clusters distances and the normalized spatial mean of the corresponding secondary large-scale field.

Once the downscaling process is done, the resampling is performed, e.g. the algorithm finds the analog days given the normalized clusters distances, the normalized total precipitation calculated using regression, and the normalized spatial mean of

Listing 5: Example NetCDF file convention for non-gridded observation data

```
netcdf RR_1d_19850801_19860731 {
dimensions:
        lon = 12 ;
        lat = 12 ;
        time = UNLIMITED ; // (365 currently)
variables:
        float lat(lon) ;
                lat:units = "degrees_north" ;
                lat:long_name = "latitude coordinate" ;
                lat:standard_name = "latitude" ;
        float lon(lat) ;
                lon:units = "none" ;
                lon:long_name = "longitude coordinate" ;
                lon:standard_name = "longitude" ;
        float alt(lon) ;
                alt:units = "meters" ;
                alt:long_name = "Altitude" ;
                alt:standard_name = "Altitude" ;
                alt:coordinates = "lon lat" ;
                alt:missing_value = -99s ;
                alt:grid_mapping = "list" ;
        int time(time) ;
                time:units = "hours since 1900-01-01T00:00:00Z" ;
                time:long_name = "time in hours since 1900-01-01T00:00:00Z" ;
                time:standard = "time in hours since 1900-01-01T00:00:00Z" ;
                time:calendar = "gregorian" ;
        float RR(time, lon) ;
                RR:units = "mm" ;
                RR:long_name = "Total precipitation" ;
                RR:standard_name = "prtot" ;
                RR:coordinates = "lon" ;
                RR:missing_value = -999.f ;
                RR:grid_mapping = "list" ;
}
```

the secondary large-scale field. The search is performed for each season separately, and then for each downscaled day. There are several configuration parameters that influence the search.

2.4 Output process

Once the analog days are determined, it is just a matter of reading observation data for each analog day of each downscaled day, for each downscaled period. A temperature correction is performed and relevant observation variables are corrected when the temperature difference between the climate model downscaled day and the re-analysis analog day is greater than a specified threshold.

The observation database used can be different than the one used in the learning process. It can be either a gridded observation database or just a time series of individual data points. However, each file must contain only one year.

If the analog dates were previously saved, it is possible to run dsclim by using only the observation database and write the downscaled output. This can be a huge disk space saver as only the observation database and analog dates (along with temperature correction) are needed.

3 Building blocks

The algorithm structure has been carefully designed. Several building blocks, or stand-alone basic subroutines and functions have been designed. They can be used outside the dsclim algorithm, which is a nice feature. This has already been done with the Michelangeli classification algorithm. These building blocks have been separated in several distinct libraries which will be discussed in the next paragraphs.

3.1 Misc library

The misc library only contains two subroutines. One is alloc_error which is used to dump the memory core in case of memory allocation problems. The other one is banner, which is simply used to present a banner when a program begins and ends, along with date and time as well as consumed CPU time.

3.2 Utils library

The utils library is for general purpose subroutines and functions:

- Allocate memory for different variable types using virtual memory
- Convert non-gregorian calendar data to gregorian
- Convert NetCDF udunits time values to year, month, day, hour, minutes, seconds
- Change the date of origin for udunits time
- Compute the mean and variance of a spatially-averaged field
- Compute a spatial average of a field
- Extract a subsection of a field which covers a common time period between two time vectors
- Extract a spatial subdomain of a field
- Extract a time sub-period of a field given specified months (season extraction)
- Mask a field using a domain bounding box

- Normalize a field given its mean and variance
- Compute the distance between two latitude-longitude points
- Find a string value in a vector of strings

3.3 Filter library

The filter library contains a subroutine to apply a filter using a window on a 3D field, the third dimension being time. The only currently supported filter type is hanning using wrap edges.

3.4 Clim library

The clim library is dedicated to specific climatology utilities:

- Compute daily climatology for climatological months of a daily 3D field, the third dimension being time
- Remove seasonal cycle of a daily 3D field using a time filter
- Compute the day of year in a 366-day climatological year given a day and a month

3.5 Regress library

The regress library contains two simple subroutines. The first one is to compute regression coefficients and constant using two vectors, and the second one is to apply regression coefficients and constant to a 2D field.

3.6 XML_utils library

The xml_utils library contains subroutines useful to read and extract values in an XML file, using XPath.

3.7 PCEOF library

The pceof library is dedicated to Principal-Components and EOF subroutines. The first one normalize a 2D variable by the norm of the first column of the first dimension and it recomputes the new norm. The second one project a 3D field onto EOFs.

3.8 Classif library

The classif library implements a generic Michelangeli classification algorithm with the following subroutines:

- Output a vector (dimension days) containing the closer (Euclidian distance) cluster number. The distance is computed as the distance between a day's principal components for each EOF and the cluster principal components for each EOF. To evaluate the closest one, all the square of the distances for all EOFs are summed for each cluster, before the square root is applied.
- Algorithm to generate clusters based on the Michelangeli et al. (1995) methodology
- Algorithm to generate best clusters among many tries
- Compute mean and variance of distances to clusters
- Compute distances to clusters normalized by control run mean and variance

3.9 IO library

The io library contains lots of basic high-level subroutines to access and write data from NetCDF files:

- Read dimensions in a NetCDF file for 3D variables
- Read latitude and longitude dimensions in a NetCDF file
- Read X and Y dimensions in a NetCDF file
- Read dimensions in a NetCDF file for EOF data
- Read a 3D variable in a NetCDF file, and return information in special data structures

- Read a 2D field from a 3D variable in a NetCDF file, and return information in special data structures
- Read a 2D variable in a NetCDF file, and return information in special data structures
- Read a 1D variable in a NetCDF file, and return information in special data structures
- Read a NetCDF variable scalar double at the index position, and return information special data structures
- Write a 3D field in a NetCDF output file
- Write a 2D field in a 3D NetCDF variable
- Write NetCDF dimensions and create a NetCDF output file
- Create a new NetCDF file with global CF-1.0 attributes
- Get main time attributes in a NetCDF file
- Get NetCDF string attribute
- Get time information in a NetCDF file
- Compute time info from NetCDF time
- Handle NetCDF error

4 Configuration parameters

The dsclim algorithm uses an XML configuration file, along with default values for most of the parameters. Some parameters are required, and if they are not provided in the configuration file, the program aborts. It is also the case if there is a syntax error in the XML syntax. Non-tested options for non-default values are identified. One must be **very** careful with these configuration parameters, because the dsclim algorithm has no mean to verify many important interactions between these parameters, because of the very limited standards in the NetCDF file format and file naming protocol. By example, if providing learning data pre-calculated, one must be sure that it match the model and time period in the several input files, etc.

4.1 General settings

Debugging output. *String value: On or Off.*

```
<setting name="debug">On</setting>
```

Format of NetCDF output. *Integer value, either 3 or 4 for NetCDF-3 or NetCDF-4.* NetCDF-4 is only available if compiled using NetCDF-4 library with –enable-netcdf4 configure argument. If using NetCDF-4, output format is followign the Classic Model of NetCDF-3 files so it should be readable by standard utilities and libraries.

```
<setting name="format">4</setting>
```

Compression of output files when using NetCDF-4 format. *String value On or Off.* Beware that some utilities, if not updated to use NetCDF-4, will not be able to read compressed files.

```
<setting name="compression">Off</setting>
```

If we want to only output downscaled data using pre-calculated analog days. *Integer value 0 or 1.*

```
<setting name="output_only">0</setting>
```

If we want to save analog dates and delta of temperature files. *Integer value 0 or 1.*

```
<setting name="analog_save">1</setting>
```

Analog dates and delta of temperature files for control period if downscaled. *String value.*

```
<setting name="analog_file_ctrl">analog_1950_1999_EB2.nc</setting>
```

Analog dates and delta of temperature files for downscaled period. *String value.*

```
<setting name="analog_file_other">analog_2000_2050_EB2.nc</setting>
```

Climatology filter width in days. *Integer value.* **Be careful when modifying this parameter.**

```
<setting name="clim_filter_width">60</setting>
```

Climatology filter type. *String value.* Only *hanning* is supported for now.

```
<setting name="clim_filter_type">hanning</setting>
```

Cluster classification distance type. *String value.* Only *euclidian* is supported for now.

```
<setting name="classif_type">euclidian</setting>
```

Number of different classifications to try when choosing the best classification. *Integer value.* **Be careful when modifying this parameter.**

```
<setting name="number_of_partitions">50</setting>
```

Maximum number of iteration in the classification scheme when converging. *Integer value.* **Be careful when modifying this parameter.**

```
<setting name="number_of_classifications">1000</setting>
```

Base udunits time for output. *String value.* It should be a units of days or smaller. **Be careful when modifying this parameter.**

```
<setting name="base_time_units">hours since 1900-01-01 00:00:00</setting>
```

Base udunits calendar type for output. *String value.* Only *gregorian* is supported for now.

```
<setting name="base_calendar_type">gregorian</setting>
```

Longitude dimension name for EOF NetCDF data files. *String value.*

```
<setting name="longitude_name_eof">lon</setting>
```

Latitude dimension name for EOF NetCDF data files. *String value.*

```
<setting name="latitude_name_eof">lat</setting>
```

EOF dimension name for EOF NetCDF data files. *String value.*

```
<setting name="eof_name">eof</setting>
```

Points dimension name for learning NetCDF data files. *String value.*

```
<setting name="pts_name">pts</setting>
```

Cluster dimension name for learning NetCDF data files. *String value.*

```
<setting name="clust_name">clust</setting>
```

Large-scale domain bounding box used for classification (Large-Scale Circulation). *Float values.* **Be careful when modifying this parameter.**

```
<setting name="domain_large_scale">
  <longitude type="min">-10.0</longitude>
  <longitude type="max">17.5</longitude>
  <latitude type="min">35.0</latitude>
  <latitude type="max">57.5</latitude>
</setting>
```

Large-scale domain bounding box for secondary large-scale fields. *Float values.* A different domain bounding box can be specified for secondary large-scale fields (e.g. temperature). **Be careful when modifying this parameter.**

```
<setting name="domain_secondary_large_scale">
  <longitude type="min">-10.0</longitude>
  <longitude type="max">17.5</longitude>
  <latitude type="min">35.0</latitude>
  <latitude type="max">57.5</latitude>
</setting>
```

Learning mask domain bounding box. *Float values.* Optional region to mask rectangular domain when performing learning process. In this example below, Corsica is excluded. Use -999 if you don't want to have a learning mask. **Be careful when modifying this parameter. Don't forget to modify it if you change the domain!**

```
<setting name="domain_learning_mask">
  <longitude type="min">8.5</longitude>
  <longitude type="max">43.5</longitude>
  <latitude type="min">20.0</latitude>
  <latitude type="max">60.0</latitude>
</setting>
```

Output directory, output timestep, month in which yearly output files begin, and global NetCDF meta-data for output files. *The parameter timestep is important because it is the timestep used for the downscaled output. Only daily is supported for now. The month_begin option has not been tested yet!*

```
<setting name="output">
  <path>/downscaling/data/results/SRESA1B</path>
  <month_begin>08</month_begin>
  <timestep>daily</timestep>
  <title>Downscaling data from Cerfacs</title>
  <title_french>Donnees de desagregation produites par le Cerfacs</title_french>
  <summary>Downscaling data from Cerfacs</summary>
  <summary_french>Donnees de desagregation produites par le Cerfacs</summary_french>
  <description>Downscaling data from Cerfacs</description>
  <keywords>climat,scenarios,desagregation,downscaling,Cerfacs</keywords>
  <processor>C programming language</processor>
  <institution>Cerfacs</institution>
  <creator_email>globc@cerfacs.fr</creator_email>
  <creator_url>http://www.cerfacs.fr/globc/</creator_url>
  <creator_name>Global Change Team</creator_name>
  <version>1.0</version>
  <scenario>SRESA1B</scenario>
  <scenario_co2>A1B</scenario_co2>
  <model>ARPEGE grille etiree</model>
  <institution_model>Meteo-France CNRM/GMGEC</institution_model>
  <version>1.0</version>
  <country>France</country>
  <member>1</member>
  <downscaling_forcing>SAFRAN 1981-2005</downscaling_forcing>
  <contact_email>christian.page@cerfacs.fr</contact_email>
  <contact_name>Christian PAGE</contact_name>
  <other_contact_email>laurent.terray@cerfacs.fr</other_contact_email>
  <other_contact_name>Laurent TERRAY</version>
</setting>
```

4.2 Observation database settings

The observation database settings are in a block of the following tag:

```
<setting name="observations">
</setting>
```

Inside this observations tag, here are the configuration parameters available.

X and Y dimension names. *String value.*

```
<dimx_name>x</dimx_name>
<dimy_name>y</dimy_name>
```

Dimension coordinates in 1D or 2D. *String value.*

```
<dim_coordinates>1D</dim_coordinates>
```

Longitude dimension name for observations NetCDF data files. *String value.*

```
<longitude_name>lon</longitude_name>
```

Latitude dimension name for observations NetCDF data files. *String value.*

```
<latitude_name>lat</latitude_name>
```

Observation variable dimensions in 1D or 2D. *String value.*

```
<coordinates>2D</coordinates>
```

Time dimension name for observations NetCDF data files. *String value.*

```
<time_name>time</time_name>
```

Number of observations variables. *Integer value.*

```
<number_of_variables>12</number_of_variables>
```

Frequency of observation data. *String value.* Only daily is supported for now.

```
<frequency>daily</frequency>
```

Path for observation NetCDF data files. *String value.*

```
<path>/contrex/Obs/SAFRAN/netcdf</path>
```

The month in which yearly observation NetCDF data files begin. *Integer value. Not tested yet!*

```
<month_begin>08</month_begin>
```

The number of digits used to represent years in observation NetCDF data filenames. *String value.*

```
<year_digits>2</year_digits>
```

The altitude NetCDF filename for the observation database. *String value.*

```
<altitude>safran_altitude.nc</altitude>
```

The altitude NetCDF variable name for the observation database. *String value.*

```
<altitude_name>Altitude</altitude_name>
```

The format template for observation NetCDF data filenames. *String value.*

```
<!-- ForcT.DAT_france_0102_daily.nc : format as in sprintf -->
<!-- Must be consistent with the number of year_digits and month_begin. -->
<!-- If month_begin is 1, only one %d must appear! -->
<template>Forc%s.DAT_france_%02d%02d_daily.nc</template>
```

Description of the observation variables. id is the count number of the variable (beginning with id=1). acronym is the name of the variable in the NetCDF file. netcdfname is the CF-1.0 compliant NetCDF name for the variable. factor is the scale factor to apply when reading data. delta is the delta to apply when reading data. Inside the name tag, there is the CF-1.0 compliant description of the variable. The postprocess flag (yes or no value) is used to specify if the variable will be read from the observation database or calculated. Available posprocess variables are all shown below. When specifying a postprocess variable, all the variables used as input to calculate the specified variable must be present in the list. Also, the first variable in the list must be a non-postprocessed variable. *Please note that long lines are broken.*

```
<variables>
   <name id="1" acronym="T" netcdfname="tas" factor="1.0" delta="0.0" postprocess="no"
                 units="K" height="2m">Temperature at 2 m</name>
   <name id="2" acronym="Q" netcdfname="hus" factor="1.0" delta="0.0" postprocess="no"
                 units="kg kg-1" height="2m">Specific humidity at 2 m</name>
   <name id="3" acronym="PRCP" netcdfname="prr" factor="1.0" delta="0.0" postprocess="no"
                 units="kg m-2 s-1" height="surface">Liquid precipitation at the surface</name>
   <name id="4" acronym="SNOW" netcdfname="prsn" factor="1.0" delta="0.0" postprocess="no"
                 units="kg m-2 s-1" height="surface">Solid precipitation at the surface</name>
   <name id="5" acronym="RAT" netcdfname="rlds" factor="1.0" delta="0.0" postprocess="no"
                 units="W m-2" height="surface">Incoming infra-red radiation at the surface</name>
   <name id="6" acronym="GLO" netcdfname="rsds" factor="1.0" delta="0.0" postprocess="no"
                 units="W m-2" height="surface">Incoming shortwave radiation at the surface</name>
   <name id="7" acronym="Vu" netcdfname="uvas" factor="1.0" delta="0.0" postprocess="no"
                 units="m s-1" height="10m">Wind speed module at 10 m</name>
   <name id="8" acronym="TX" netcdfname="tasmax" factor="1.0" delta="0.0" postprocess="no"
                 height="2m">Maximum Daily Temperature at 2 m</name>
   <name id="9" acronym="TN" netcdfname="tasmin" factor="1.0" delta="0.0" postprocess="no"
                 height="2m">Minimum Daily Temperature at 2 m</name>
   <name id="10" acronym="hur" netcdfname="hur" factor="1.0" delta="0.0" postprocess="yes"
                 units="%" height="2m">Relative Humidity at 2 m</name>
   <name id="11" acronym="evapn" netcdfname="evapn" factor="1.0" delta="0.0" postprocess="yes"
                 units="kg m-2 s-1" height="2m">Potential Evapotranspiration at 2 m</name>
   <name id="12" acronym="prtot" netcdfname="prtot" factor="1.0" delta="0.0" postprocess="yes"
                 units="kg m-2 s-1" height="surface">Total precipitation at the surface</name>
</variables>
```

4.3 Learning settings

The learning settings are in a block of the following tag:

```
<setting name="learning">
</setting>
```

Inside this learning tag, here are the configuration parameters available.

If learning data is provided in NetCDF input files. *Integer value 0 or 1.*

```
<learning_provided>0</learning_provided>
```

If learning data is saved in NetCDF output files. *Integer value 0 or 1.*

```
<learning_save>1</learning_save>
```

NetCDF filename for input learning weights. *String value.*

```
<filename_open_weight>/data/Poid_down.nc</filename_open_weight>
```

NetCDF filename for input learning data. *String value.*

```
<filename_open_learn>/data/learning_data_NCEP.nc</filename_open_learn>
```

NetCDF filename for input learning cluster data. *String value.*

```
<filename_open_clust_learn>/data/clust_learn.nc</filename_open_clust_learn>
```

NetCDF filename for output learning weights. *String value.*

```
<filename_save_weight>/data/Poid_down.nc</filename_save_weight>
```

NetCDF filename for output learning data. *String value.*

```
<filename_save_learn>/data/learning_data_NCEP.nc</filename_save_learn>
```

NetCDF filename for output learning cluster data. *String value.*

```
<filename_save_clust_learn>/data/clust_learn.nc</filename_save_clust_learn>
```

NetCDF filename for input EOF for observed data. *String value.* Only EOF of precipitation observation is supported for now.

```
<filename_obs_eof>/data/PRE_8105_aseason_EOF.nc</filename_obs_eof>
```

NetCDF filename for input EOF of circulation large-scale field reanalysis data. *String value.*

```
<filename_rea_eof>/data/psl_1d_19480101_20060331_NCP_EOF.nc</filename_rea_eof>
```

NetCDF filename for input EOF of secondary large-scale field reanalysis data. *String value.*

```
<filename_rea_sup>/data/tas_1d_19480101_20070331_NCP.nc</filename_rea_sup>
```

Number of EOFs for observation learning data. *Integer value.* **Be careful when modifying this parameter.**

```
<number_of_obs_eofs>10</number_of_obs_eofs>
```

NetCDF variable name for EOF observation data. *String value.*

```
<nomvar_obs_eof>pre_pc</nomvar_obs_eof>
```

NetCDF variable name for Singular Values of observation data. *String value.*

```
<nomvar_obs_sing>pre_sing</nomvar_obs_sing>
```

Number of EOFs for circulation large-scale field reanalysis. *Integer value.* **Be careful when modifying this parameter.**

```
<number_of_rea_eofs>10</number_of_rea_eofs>
```

NetCDF variable name for EOF of circulation large-scale field reanalysis. *String value.*

```
<nomvar_rea_eof>psl_pc</nomvar_rea_eof>
```

NetCDF variable name for Singular Values of circulation large-scale field reanalysis. *String value.*

```
<nomvar_rea_sing>psl_sing</nomvar_rea_sing>
```

4.4 Regression settings

The regression settings are in a block of the following tag:

```
<setting name="regression">
</setting>
```

Inside this regression tag, here are the configuration parameters available.

Regression latitude-longitude points NetCDF input filename. *String value.*

```
<filename>/data/reg_pts_france.nc</filename>
```

Longitude dimension name for NetCDF input file. *String value.*

```
<longitude_name>lon</longitude_name>
```

Latitude dimension name for NetCDF input file. *String value.*

```
<latitude_name>lat</latitude_name>
```

X and Y dimension names. *String value.*

```
<dimx_name>x</dimx_name>
<dimy_name>y</dimy_name>
```

Points dimension name for NetCDF input file. *String value.*

```
<pts_name>pts</pts_name>
```

Regression distance in meters for spatial mean. *Float value.* **Be careful when modifying this parameter.**

```
<distance>40000.0</distance>
```

If we want output of regression data (downscaling phase) for diagnostics. *Integer value 0 or 1.*

```
<regression_save>1</regression_save>
```

NetCDF filename for output of regression data (downscaling phase) for diagnostics, for the control period if downscaled. *String value.*

```
<filename_save_ctrl_reg>/data/downscaling_diagnostics_1950_1999_EB2.nc</filename_save_ctrl_reg>
```

NetCDF filename for output of regression data (downscaling phase) for diagnostics, for the downscaled period. *String value.*

```
<filename_save_other_reg>/data/downscaling_diagnostics_2000_2049_EA2.nc</filename_save_other_reg>
```

Time dimension name for observations NetCDF data files. *String value.*

```
<time_name>time</time_name>
```

4.5 Seasons settings

The seasons settings have the two following parameter which identify the number of seasons: **Number of seasons.** *Integer value.*

```
<setting name="number_of_seasons">4</setting>
```

After this parameter, the following main tag is used:

```
<setting name="seasons">
</setting>
```

Inside this main tag, the following parameters are available, each season being identified with an id attribute beginning with id=1. One must be very careful when modifying these values.

For each season, number of months followed by month numbers in that particular season. *Integer list.* The nmonths attribute is the number of months listed in the list value. **Be careful when modifying this parameter.**

```
<season id="1" nmonths="3">09 10 11</season>
<season id="2" nmonths="3">12 01 02</season>
<season id="3" nmonths="3">03 04 05</season>
<season id="4" nmonths="3">06 07 08</season>
```

For each season, number of clusters. *Integer value.* **Be careful when modifying this parameter.**

```
<number_of_clusters id="1">9</number_of_clusters>
<number_of_clusters id="2">9</number_of_clusters>
<number_of_clusters id="3">10</number_of_clusters>
<number_of_clusters id="4">10</number_of_clusters>
```

For each season, number of regression variables. *Integer value.* **Be careful when modifying this parameter.**

```
<number_of_regression_vars id="1">9</number_of_regression_vars>
<number_of_regression_vars id="2">9</number_of_regression_vars>
<number_of_regression_vars id="3">10</number_of_regression_vars>
<number_of_regression_vars id="4">11</number_of_regression_vars>
```

For each season, number of ±days to search for analog day in the season starting from the downscaled day of year. *Integer value.* **Be careful when modifying this parameter.**

```
<number_of_days_search id="1">10</number_of_days_search>
<number_of_days_search id="2">10</number_of_days_search>
<number_of_days_search id="3">10</number_of_days_search>
<number_of_days_search id="4">10</number_of_days_search>
```

For each season, number of days to choose when doing the first selection of the analog day. *Integer value.* **Be careful when modifying this parameter.**

```
<number_of_days_choices id="1">16</number_of_days_choices>
<number_of_days_choices id="2">16</number_of_days_choices>
<number_of_days_choices id="3">11</number_of_days_choices>
<number_of_days_choices id="4">11</number_of_days_choices>
```

For each season, if we want to shuffle when choosing the analog day. *Integer value 0 or 1.* **Be careful when modifying this parameter.**

```
<days_shuffle id="1">1</days_shuffle>
<days_shuffle id="2">1</days_shuffle>
<days_shuffle id="3">0</days_shuffle>
<days_shuffle id="4">0</days_shuffle>
```

For each season, if we want to use the secondary large-scale field in the final selection of the analog day. *Integer value 0 or 1.* **Be careful when modifying this parameter.**

```
<secondary_field_choice id="1">0</secondary_field_choice>
<secondary_field_choice id="2">0</secondary_field_choice>
<secondary_field_choice id="3">1</secondary_field_choice>
<secondary_field_choice id="4">1</secondary_field_choice>
```

For each season, if we want to use the secondary large-scale field in the first selection of the analog day. *Integer value 0 or 1.* **Be careful when modifying this parameter.**

```
<secondary_field_main_choice id="1">1</secondary_field_main_choice>
<secondary_field_main_choice id="2">1</secondary_field_main_choice>
<secondary_field_main_choice id="3">0</secondary_field_main_choice>
<secondary_field_main_choice id="4">0</secondary_field_main_choice>
```

4.6 Control period settings (optionally downscaled)

If the control period is downscaled or not. *Integer value 0 or 1.*

```
<downscale>1</downscale>
```

The control period settings are in a block of the following tag:

```
<setting name="period_ctrl">
</setting>
```

Inside this period_control tag, here are the configuration parameters available.

Beginning date (year) for control period for downscaled output. *Integer value.*

```
<year_begin>1950</year_begin>
```

Beginning date (month) for control period for downscaled output. *Integer value.*

```
<month_begin>01</month_begin>
```

Beginning date (day) for control period for downscaled output. *Integer value.*

```
<day_begin>01</day_begin>
```

End date (year) for control period for downscaled output. *Integer value.*

```
<year_end>1999</year_end>
```

End date (month) for control period for downscaled output. *Integer value.*

```
<month_end>12</month_end>
```

End date (day) for control period for downscaled output. *Integer value.*

```
<day_end>31</day_end>
```

4.7 Downscaled period settings

The period settings are in a block of the following tag:

```
<setting name="period">
</setting>
```

Inside this period tag, here are the configuration parameters available.

Beginning date (year) for downscaled period output. *Integer value.*

```
<year_begin>1950</year_begin>
```

Beginning date (month) for downscaled period output. *Integer value.*

```
<month_begin>01</month_begin>
```

Beginning date (day) for downscaled period output. *Integer value.*

```
<day_begin>01</day_begin>
```

End date (year) for downscaled period output. *Integer value.*

```
<year_end>1999</year_end>
```

End date (month) for downscaled period output. *Integer value.*

```
<month_end>12</month_end>
```

End date (day) for downscaled period output. *Integer value.*

```
<day_end>31</day_end>
```

4.8 Large-scale fields settings

Large-scale field settings have the two following parameters which identify the number of fields for both the control and the downscaled period:

Number of circulation large-scale fields for control period. *Integer value.*

```
<setting name="number_of_large_scale_control_fields">1</setting>
```

Number of circulation large-scale fields for downscaled period. *Integer value.*

```
<setting name="number_of_large_scale_fields">1</setting>
```

After these parameters for large-scale control fields, the following main tag is used:

```
<setting name="large_scale_control_fields">
</setting>
```

After these parameters for large-scale fields (downscaled period), the following main tag is used:

```
<setting name="large_scale_fields">
</setting>
```

Inside these main tags, there are many configuration parameters available describing each large-scale field. Each tag has an id attribute (beginning with id=1) to specify the corresponding variable. Currently, only 1 circulation large-scale field is allowed, and only one downscaled period.

NetCDF circulation large-scale variable name. *String value.*

```
<name id="1">psl</name>
```

NetCDF input filename. *String value.*

```
<filename id="1">/data/interpERA40_psl_1d_19500101_19991231_EB2.nc</filename>
```

NetCDF projection type CF-1.0 compliant. *String value.* Only Latitude_Longitude and Lambert_Conformal are supported.

```
<projection id="1">Latitude_Longitude</projection>
```

Number of dimensions for coordinates variables in NetCDF input file. *String value.* Values of 1D or 2D are supported.

```
<coordinates id="1">2D</coordinates>
```

Longitude dimension name for NetCDF data files. *String value.*

```
<longitude_name>lon</longitude_name>
```

Latitude dimension name for NetCDF data files. *String value.*

```
<latitude_name>lat</latitude_name>
```

Time dimension name for NetCDF data files. *String value.*

```
<time_name>time</time_name>
```

If we need to remove climatology or not. *Integer value 0 or 1.*

```
<clim_remove id="1">1</clim_remove>
```

NetCDF climatology variable name. *String value.*

```
<clim_name id="1">psl</clim_name>
```

If climatology is already provided or not. *Integer value 0 or 1.*

```
<clim_provided id="1">0</clim_provided>
```

If we want to save climatology or not in an output NetCDF file. *Integer value 0 or 1.*

```
<clim_save id="1">0</clim_save>
```

NetCDF file to open for pre-calculated climatology. *String value.*

```
<clim_openfilename id="1">/data/CLIM_psl_1d_19500101_19991231.nc</clim_openfilename>
```

NetCDF file to save climatology. *String value.*

```
<clim_savefilename id="1">/data/CLIM_psl_1d_19500101_19991231_save.nc</clim_savefilename>
```

Project field onto EOF or not. *Integer value 0 or 1.* **Be careful when modifying this parameter.**

```
<eof_project id="1">1</eof_project>
```

Number of EOFs. *Integer value.* **Be careful when modifying this parameter.**

```
<number_of_eofs id="1">10</number_of_eofs>
```

Scaling factor when projecting data onto EOF. *Float value.* **It is very important that this parameter be correctly adjusted!**

```
<eof_scale id="1">100.0</eof_scale>
```

NetCDF variable name for projected field onto EOF. *String value.* Useful only when reading or saving EOF data.

```
<eof_name id="1">psl_eof</eof_name>
```

NetCDF variable name for singular values. *String value.* Useful only when reading or saving EOF data.

```
<sing_name id="1">psl_sing</sing_name>
```

Number of dimensions for coordinates variables in NetCDF EOF input file. *String value.* Values of 1D or 2D are supported.

```
<eof_coordinates id="1">1D</eof_coordinates>
```

NetCDF input file for pre-projected data on EOF. *String value.* Useful only when reading EOF data.

```
<eof_openfilename id="1">/data/psl_1d_19480101_20060331_NCP_aseason_EOF.nc</eof_openfilename>
```

4.9 Secondary large-scale fields settings

Seconary large-scale field settings have the two following parameters which identify the number of fields for both the control and the downscaled period:

Number of secondary large-scale fields for control period. *Integer value.*

```
<setting name="number_of_secondary_large_scale_control_fields">1</setting>
```

Number of secondary large-scale fields for downscaled period. *Integer value.*

```
<setting name="number_of_secondary_large_scale_fields">0</setting>
```

For secondary large-scale control fields, there is an optional mask file that can be used. However, this mask must be on the same grid as the secondary large scale field data.

```
<setting name="domain_secondary_large_scale_mask">
</setting>
```

Inside this main tag, the following configuration parameters are available.

Use optional mask NetCDF file. *Integer value 0 or 1.*

```
<use_mask>0</use_mask>
```

Mask NetCDF filename. *String value.*

```
<filename>/data/secondary_large_scale_mask.nc</filename>
```

Mask NetCDF variable name. *String value.*

```
<mask_name>mask</mask_name>
```

Longitude dimension name for mask NetCDF data file. *String value.*

```
<longitude_name>lon</longitude_name>
```

Latitude dimension name for mask NetCDF data file. *String value.*

```
<latitude_name>lat</latitude_name>
```

X and Y dimension names. *String value.*

```
<dimx_name>lon</dimx_name>
<dimy_name>lat</dimy_name>
```

Number of dimensions for coordinates variables in NetCDF mask file. *String value.* Values of 1D or 2D are supported.

```
<coordinates id="1">2D</coordinates>
```

Dimension coordinates in 1D or 2D. *String value.*

```
<dim_coordinates>1D</dim_coordinates>
```

NetCDF projection type CF-1.0 compliant. *String value.* Only Latitude_Longitude is supported.

```
<projection id="1">Latitude_Longitude</projection>
```

For secondary large-scale control fields, the following main tag is used:

```
<setting name="secondary_large_scale_control_fields">
</setting>
```

For secondary large-scale fields (downscaled period), the following main tag is used:

```
<setting name="secondary_large_scale_fields">
</setting>
```

Inside these main tags, there are many configuration parameters available describing each secondary large-scale field. Each tag has an id attribute (beginning with id=1) to specify the corresponding variable. Currently, only 1 secondary large-scale field is allowed, and a temperature field is assumed. Also, only one downscaled period is allowed.

NetCDF secondary large-scale variable name. *String value.*

```
<name id="1">tas</name>
```

NetCDF input filename. *String value.*

```
<filename id="1">/data/interpERA40_tas_1d_19500101_19991231_EB2_test.nc</filename>
```

NetCDF projection type CF-1.0 compliant. *String value.* Only Latitude_Longitude and Lambert_Conformal are supported.

```
<projection id="1">Latitude_Longitude</projection>
```

Number of dimensions for coordinates variables in NetCDF input file. *String value.* Values of 1D or 2D are supported.

```
<coordinates id="1">2D</coordinates>
```

Longitude dimension name for NetCDF data files. *String value.*

```
<longitude_name>lon</longitude_name>
```

Latitude dimension name for NetCDF data files. *String value.*

```
<latitude_name>lat</latitude_name>
```

Time dimension name for NetCDF data files. *String value.*

```
<time_name>time</time_name>
```

5 Compiling and installing

The dsclim software package is meant to be easy to compile and install on recent unix platforms. It is built using the standard GNU autoconf/automake system. It is available here: http://www.cerfacs.fr/ page/dsclim/dsclim-1.x.x.tar.gz where 1.x.x is the version number. At the time of release of this documentation, the version number is 1.0.4. It is available under the CeCILL open-source license (CEA - CNRS - INRIA, 2008).

The requirements are the following:

- A 32- or 64-bit recent unix system (tested on Linux 32- and 64-bit)
- The gzip, bzip2 and tar utilities
- A recent ANSI C compiler (tested with GNU GCC 4.x)
- A standard C math library (libm)
- The standard zlib data compression library (libz)
  http://www.zlib.net/
- The standard szip compression library (libsz)
  ftp://ftp.hdfgroup.org/lib-external/szip/
- The HDF5 library (libhdf5)
  ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf-4/
  and http://www.hdfgroup.org/HDF5/
- UCAR NetCDF 4.x (recommended) or 3.6.3 library (libnetcdf)
  http://www.unidata.ucar.edu/software/netcdf/
- UCAR UDUNITS 1.x library (libudunits)
  http://www.unidata.ucar.edu/udunits/
- The standard GNU Scientific Library (libgsl)
  http://www.gnu.org/software/gsl/
- The standard XML2 library (libxml2)
  http://xmlsoft.org/

Most of these should be already available on recent Linux systems, or directly available through a packaging system like yum or urpmi.

Once the archive is uncompressed and untarred using

```
tar xvzf dsclim-1.x.x.tar.gz
```

one should run the *configure* utility inside the dsclim-x.x.x directory. Here is one example when one want to install in a different directory than /usr/local/bin and when needed libraries have been installed in non-standard directories:

```
./configure --prefix=/usr/local/dsclim \\
--with-zlib=/usr/local64 \\
--with-szlib=/usr/local64 \\
--with-hdf5=/usr/local64 \\
--with-netcdf=/usr/local64 \\
--with-udunits=/usr/local64
```

The next step is to compile by typing the command *make*, followed by *make doxygen-doc* (only if you have doxygen installed) and *make install* if everything went fine in the *make* process to install everything. The executable is installed as *$prefix/bin/dsclim*, an example configuration file is installed as *$prefix/share/dsclim/configuration_example_grid.xml*. *$prefix* stands as the directory specified by the *prefix* argument when running the *configure* utility (*/usr/local/bin* by default).

Datafiles for an example case are provided in a separate tar archive. It is suggested to put the content of this archive in a specific directory to be referenced in the configuration file.

Once installed, it is advised to copy the example *configuration_example_grid.xml* file and modify it for your needs. Then, if the *$prefix* directory is in your executable *$PATH* variable, you can run dsclim just by typing

```
dsclim -conf configuration_example_grid.xml
```

assuming that the *configuration_example_grid.xml* file is in the current directory, and that you have all the input datafiles available at the proper locations.

6 Examples

This section will present a few examples on the use of the dsclim downscaling software. The first one will show how to generate learning data and downscale a climate model simulation. The second one will show how to use a non-gridded observation database when learning data has already been generated.

6.1 First example: generate learning data and downscale using a gridded observation database

A first example case will be used here to present how dsclim can be used to downscale climate numerical model data and generate learning data. This example uses the Meteo-France SAFRAN (Quintana-Seguí et al., 2008) analysis as an observation database, and the configuration of the algorithm match the one of Boe08 with slight modifications. The configuration file used is *configuration_example_grid.xml* provided in the software package.

The predictant chosen is the Total Liquid Precipitation while the predictors are the Mean Sea-Level pressure and the 2-meters Temperature. The geographical domain for the classification of the weather types and the calculation of a temperature index is shown in Fig. 2. The geographical domain is specified in the configuration file by a bounding box with the *domain_large_scale* configuration parameters, and a mask domain is also used to mask Corsica using the *domain_learning_mask* parameter. The regressions are calculated on 220 points as shown in Fig. 16, covering the domain of the SAFRAN analysis. The points are specified by the means of a NetCDF file *reg_pts_france.nc*.

The learning period chosen is Aug 1st, 1981 to Jul 31st, 2005. This period has been chosen in Boe08 because it was the initial SAFRAN period available when the algorithm was first designed in 2006. The re-analysis chosen is the NCEP (Kistler et al., 2001) one: it is because the ERA40 re-analysis (Uppala et al., 2005) didn't span all the SAFRAN period at the time the study was done. An important point is that because the SAFRAN analysis is aimed at hydrology, the data is organized in hydrological years beginning August 1st. In this example, learning data is generated and saved into NetCDF files for further uses.

The period used as the control is automatically selected by the algorithm as the common period between the numerical climate model period of the NetCDF input file specified in the *large_scale_control_fields* block in the configuration file, in the parameter *filename*, and the period of the observation database (SAFRAN in this case). In this example case, the model file spans the period Jan 1st, 1950 to Dec, 31st 1999, which makes the control period run from Aug 1st, 1981 to Dec 31st, 1999.

Four seasons are chosen: Sep Oct Nov; Dec Jan Feb; Mar Apr May; Jun Jul Aug. They are defined in the *number_of_seasons* and *seasons* configuration parameters. For fall and winter seasons we use 9 clusters (*number_of_clusters*), while we use 10 for spring and summer. A temperature index is used in the regression for the summer season only (*number_of_regression_vars* vs *number_of_clusters*). The search for analog days is ±10 days around the day of year of the downscaled date (*number_of_days_search*), and a temperature index is also used in the search for the fall and winter seasons (*secondary_field_main_choice*). For fall and winter, 16 days are chosen in the first selection, while 11 days are chosen for spring and summer (*number_of_days_choices*). Shuffling is performed for the final selection for fall and winter (*days_shuffle*), while the day having the closest temperature index is chosen for the spring and summer seasons (*secondary_field_choice*).

Climatology is removed for the Mean Sea-Level Pressure field used in the weather types classification, using a Hanning filter and a 60 days window. The classification is done using Euclidian distance, a maximum of 1000 iterations is allowed for convergence and 30 different partitions are tried (starting from random points) before choosing the best one.

The EOFs for the re-analysis Mean Sea-Level Pressure and the Total Liquid Precipitation from the SAFRAN observation database are pre-calculated using the STATPACK software package. Also, the numerical climate model output is already pre-interpolated onto the re-analysis grid using OASIS version 3 (Valcke, 2006). Seven observation variables are available in the SAFRAN observation database in NetCDF files, which are configured in the *variables* parameter of the *observations* configuration block. These variables are stored as daily data in files spanning one year each, from Aug 1st to Jul 31st:

- Temperature at 2 meters
- Specific humidity at 2 meters
- Liquid precipitation at the surface
- Solid precipitation at the surface
- Incoming infra-red radiation at the surface
- Incoming shortwave radiation at the surface
- Wind speed module at 10 meters

These variables are on an regular 8 km X 8 km grid on a Lambert-II projection as shown in Fig. 17.

For the downscaling itself, in this example, a simulation of the Meteo-France ARPEGE V4.1 climate model is used. The simulation is broken into two separate output files, one for the period 1950-2000 and the second for the period 2000-2050. The first period is used as input for the control period (*filename* in *large_scale_control_fields* and *secondary_large_scale_control_fields* blocks), and is downscaled (*downscale* is 1 in *period_ctrl* block) along with the second period (*filename* in *large_scale_fields* and *secondary_large_scale_fields* blocks). Only the Mean Sea-Level Pressure and the Temperature at 2 meters are needed from the model output. However, it must be already interpolated to the NCEP re-analysis grid, which is the grid that was used for the learning process.

The configuration is also set to save the generated analog dates along with the tem-

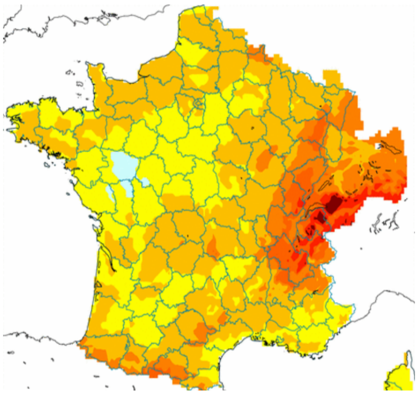

FIG. 17: The observation analysis SAFRAN geographical domain and grid.

perature change. This enables the user to regenerate the downscaling output with just this file and the observation database. However, the user must be careful to use the same configuration options that were used. For this purpose, the configuration file is also saved in the analog data output file.

To start the program, one just need to type the following command, assuming that the executable dsclim has been put into a directory path in the PATH environment variable, and that the file configuration_example_grid.xml is in the current directory:

```
dsclim -conf configuration_example_grid.xml
```

The program will then execute, output information on the terminal and write the data. It is suggested to name XML configuration files with meaningful names. If there is any error in the configuration file, the program will abort execution. Once downscaling data has been generated, it can be viewed or read using utilities which can read NetCDF files, such as NCL, IDV, ncview, IDL, etc. The NetCDF convention used is the widely standard CF-1.0.

The result of the downscaling process is thus stored as NetCDF files.

The methodology being based on observed precipitation in the learning period for regression and weather types, it performs quite well in representing the correctly the precipitation climatology and variability since the numerical model downscaled (ARPEGE V4.1) represents adequatly the weather types in the period corresponding to the NCEP re-analysis (Boé et al., 2007). The following paragraphs will show some results of the downscaling performed as described above.

The mean total precipitation anomalies (mm) over France over the period 2080-2098 are shown in Fig. 18.

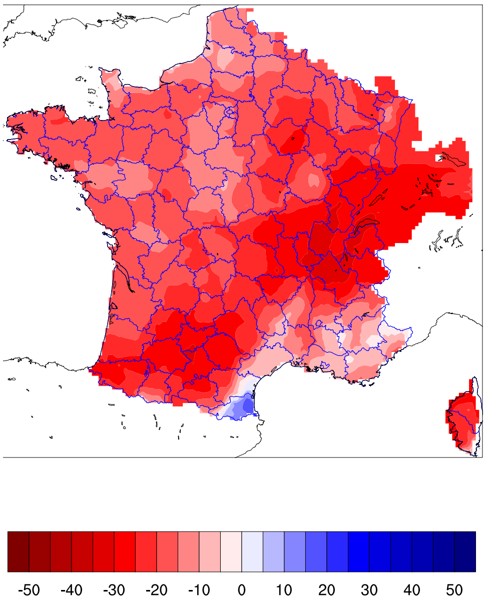

FIG. 18: Mean downscaled total precipitation anomalies (mm) over France of an ARPEGE V4.1 simulation over the period 2080-2098 (1971-1990 climatology).

As a comparison of the model downscaled data against the observation database SAFRAN, here is the mean annual total precipitation (mm) over the period 1971-1998 (Figs. 19 and 20).

The difference in percentage between the downscaled model and SAFRAN is also shown in Fig. 21, and against NCEP in Fig. 22. These figures shows that the downscaling algorithm performance is satisfactory, with mean errors on the order of 1 to 2%, and maximum errors on the order of 15%.

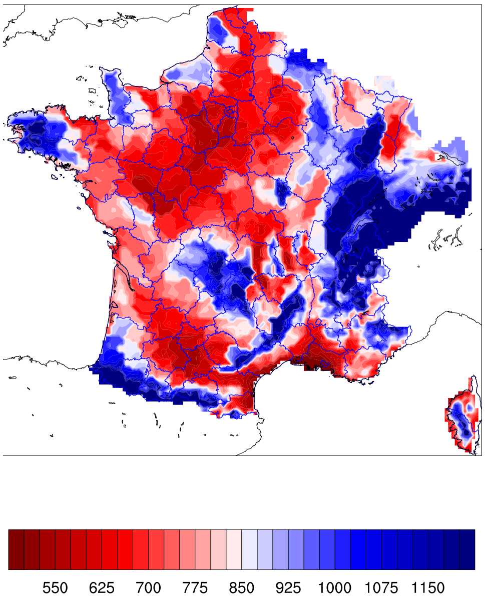

Fig. 19: Mean downscaled annual total precipitation (mm) over France of an ARPEGE V4.1 simulation over the period 1971-1998.

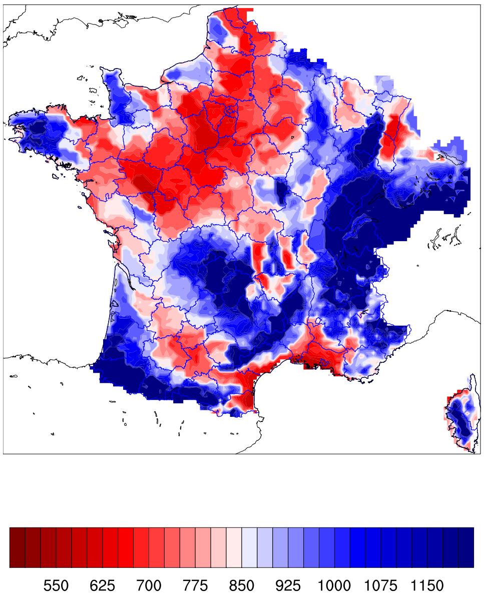

Fig. 20: Mean SAFRAN analysis annual total precipitation (mm) over France over the period 1971-1998.

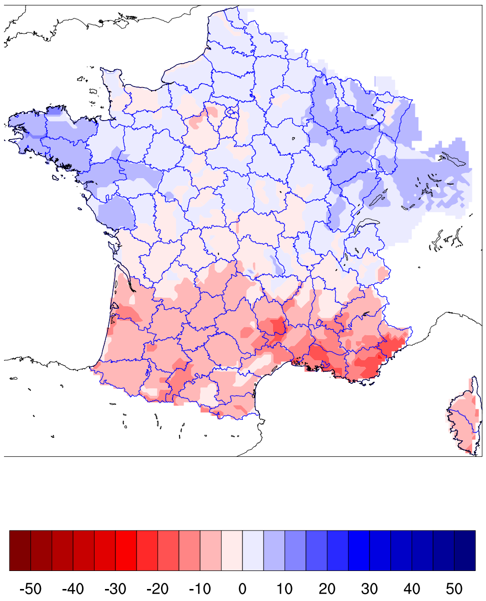

FIG. 21: Difference ARPEGE V4.1 vs SAFRAN of mean annual total precipitation (%) over France over the period 1971-1998.

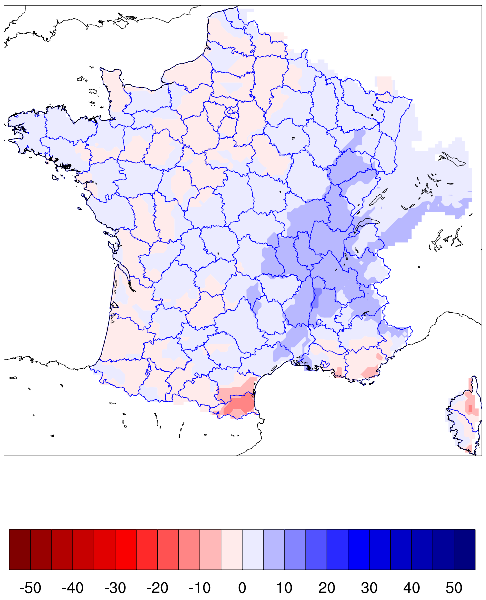

FIG. 22: Difference ARPEGE V4.1 vs NCEP of mean annual total precipitation (%) over France over the period 1971-1998.

6.2 Second example: using pre-generated learning data and downscale using a non-gridded observation database

This second example case will be used to present how dsclim can be used with a non-gridded observation database, provided that learning data has already been generated using a gridded observation database. Currently, it is not possible to use a non-gridded observation database to generate learning data.

In this example, the learning data used is the one generated and saved in the first example, using the gridded Meteo-France SAFRAN (Quintana-Seguí et al., 2008) analysis as an observation database. The configuration file used is *configuration_example_nongrid.xml* provided in the software package. The configuration is the same as in the first example, except for the description of the observation database.

The non-gridded observation database in this example comes from the CLIMATOR project. It consists of six variables, which are defined in the *variables* parameter of the *observations* configuration block. Originally the data was stored in an ASCII format, but for to interface with the dsclim algorithm, data has been stored into CF-1.0 compliant NetCDF files. These variables are stored as daily data in files spanning one year each, from Aug 1st to Jul 31st:

- Maximum Temperature at 2 m
- Minimum Temperature at 2 m
- Relative humidity at 2 m
- Total precipitation at the surface
- Incoming shortwave radiation at the surface
- Wind speed module at 10 m

For the downscaling itself, the same climate simulation has been used as in the first example, which is a simulation of the Meteo-France ARPEGE V4.1 climate model is used. As in the first example, the simulation is broken into two separate output files, one for the period 1950-2000 and the second for the period 2000-2050. The first period is used as input for the control period (*filename* in *large_scale_control_fields* and *secondary_large_scale_control_fields* blocks), and is downscaled (*downscale* is 1 in *period_ctrl* block) along with the second period (*filename* in *large_scale_fields* and *secondary_large_scale_fields* blocks). Only the Mean Sea-Level Pressure and the Temperature at 2 meters are needed from the model output. However, it must be already interpolated to the NCEP re-analysis grid, which is the grid that was used for the learning process.

The configuration is also set to save the generated analog dates along with the temperature change. This enables the user to regenerate the downscaling output with just this file and the observation database. However, the user must be careful to use the same configuration options that were used. For this purpose, the configuration file is also fully saved in the analog data NetCDF output file.

To start the program, one just need to type the following command, assuming that the executable dsclim has been put into a directory path in the PATH environment variable, and that the file configuration_example_nongrid.xml is in the current directory:

```
dsclim -conf configuration_example_nongrid.xml
```

The program will then execute, output information on the terminal and write the data. It is suggested to name XML configuration files with meaningful names. If there is any error in the configuration file, the program will abort execution. Once downscaling data has been generated, it can be viewed or read using utilities which can read NetCDF files, such as NCL, IDV, ncview, IDL, etc. The NetCDF convention used is the widely standard CF-1.0. It must be noted that the result of the downscaling process is stored on the same grid as the input data. Since the data is non-gridded, only individual time series are available.

Conclusion

The dsclim software package provides a generic, extensible, configurable and relatively easy to use implementation of a weather-type statistical downscaling methodology of climate simulations based on the innovative work of Boé (2007); Boé and Terray (2008a,b); Boé et al. (2006). The new implementation has been validated against the original coding of the methodology. The performance of the dsclim algorithm has proven to be very much better than the original coding version because it is now all coded in C language instead of using IDL.

This new software will enable end-users to downscale climate model simulations more easily. Its flexibility will enable its use in different contexts, with other observation databases and over other regions. It also provides a platform to perform sensitivity studies about the methodology itself, and to develop similar methodologies. The development of dsclim will continue, which will eventually enable faster scientific improvements of the original methodology it implements.

Bibliography

Boé, J., 2007: *Changement global et cycle hydrologique : Une étude de régionalisation sur la France*. Ph.D. thesis, Université Paul Sabatier - Toulouse III.

Boé, J. and L. Terray, 2008a: A Weather-Type Approach to Analyzing Winter Precipitation in France: Twentieth-Century Trends and the Role of Anthropogenic Forcing. *J. Climate*, **21 (13)**, 3118.

— 2008b: Régimes de temps et désagrégation d'échelle. *La Houille Blanche*, **2**, doi:10.1051/lhb:2008016L05702.

Boé, J., L. Terray, F. Habets, and E. Martin, 2006: A simple statistical-dynamical downscaling scheme based on weather types and conditional resampling. *J. Geophys. Res.*, **111**, D21106.

— 2007: Statistical and dynamical downscaling of the Seine basin climate for hydro-meteorological studies. *Int. J. Clim.*, **27**, 1643–1655.

CEA - CNRS - INRIA, 2008: Cecill.
URL `www.cecill.info`

Kistler, R., E. Kalnay, W. Collins, S. Saha, G. White, J. Woollen, M. Chelliah, W. Ebisuzaki, M. Kanamitsu, V. Kousky, H. van den Dool, R. Jenne, and M. Fiorino, 2001: The NCEP-NCAR 50-Year Reanalysis: Monthly Means CD-ROM and Documentation. *Bull. Amer. Meteor. Soc.*, **82**, 247–268.

Le Moigne, P., 2002: Description de l'analyse des champs de surface sur la France par le système SAFRAN. Technical report, Centre national de recherches météorologiques, Météo-France.

Michelangeli, P.-A., R. Vautard, and B. Legras, 1995: Weather regimes: Recurrence and quasi stationarity. *J. Atmos. Sci.*, **52 (8)**, 1237–1256.

Program for Climate Model Diagnosis and Intercomparison, 2008: CF Metadata.
URL `cf-pcmdi.llnl.gov/`

Quintana-Seguí, P., P. L. Moigne, Y. Durand, E. Martin, F. Habets, M. Baillon, C. Canellas, L. Franchisteguy, and S. Morel, 2008: Analysis of Near-Surface

Atmospheric Variables: Validation of the SAFRAN Analysis over France. *J. Appl. Met. Clim.*, **47**, 92–107.

Russell, K. R., E. J. Hartnett, and J. Caron, 2006: NetCDF-4: Software Implementing an Enhanced Data Model for the Geosciences. *22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*.

Salas y Mélia, D., F. Chauvin, M. Déqué, H. Douville, J.-F. Guérémy, P. Marquet, S. Planton, J.-F. Royer, and S. Tyteca, 2005: Description and validation of CNRM-CM3 global coupled climate model. Technical report, Centre national de recherches météorologiques, Groupe de Météorologie de Grande Echelle et Climat, Météo-France.

Uppala, S., P. Kållberg, A. Simmons, U. Andrae, V. da Costa Bechtold, M. Fiorino, J. Gibson, J. Haseler, A. Hernandez, G. Kelly, X. Li, K. Onogi, S. Saarinen, N. Sokka, R. Allan, E. Andersson, K. Arpe, M. Balmaseda, A. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, S. Caires, F. Chevallier, A. Dethof, M. Dragosavac, M. Fisher, M. Fuentes, S. Hagemann, E. Hólm, B. Hoskins, L. Isaksen, P. Janssen, R. Jenne, A. McNally, J.-F. Mahfouf, J.-J. Morcrette, N. Rayner, R. Saunders, P. Simon, A. Sterl, K. Trenberth, A. Untch, D. Vasiljevic, P. Viterbo, and J. Woollen, 2005: The ERA-40 re-analysis. *Quart. J. R. Meteorol. Soc.*, **131**.

Valcke, S., 2006: OASIS3 User Guide (prism_2-5). Technical report, PRISM Support Initiative No 3.

Vautard, R., 1990: Multiple Weather Regimes over the North Atlantic: Analysis of Precursors and Successors. *Mon. Wea. Rev.*, **118**, 2056–2081.