OR 52/2014

Norsk institutt for luftforskning
Norwegian Institute for Air Research

Background concentrations in Norway: Towards automated annual updates

Philipp Schneider

Scientific report

Contents

List of Figures

Summary

A semi-automated technique was developed for performing annual updates of the dataset on background concentrations in Norway which was produced in previous years. The code is written in the Matlab programming language and large parts of the code base are included in the Appendix of this report.

The spatial component of the system was updated to include data from 2009 through 2011. Acquiring and preparing the input data for the spatial component still requires a relatively small amount of manual effort, however the majority of the remaining process has been automated to the largest extent possible, such that only the derivation of the semivariograms for the residual kriging step requires very brief interaction by an expert user.

The temporal component has been updated to version 8 of the European air quality database (AirBase), now including several additional years up to and including 2013. Entirely new anomaly matrices have been calculated from the updated data for all background stations in Norway.

Asssuming that the availability and the format of the required input data remains unchanged, future annual updates of the system can be carried out within a very short time frame on the order of around 1-2 days.

1 Introduction

Many applications related to air quality require approximate estimates of the spatial and temporal dynamics of background concentrations of the main air pollutants. To some extent such information has already been available for several years. For example, the spatial distribution of some air pollutants is routinely mapped in an operational fashion for the European Environment Agency (EEA) by the European Topic Centre on Air Quality and Climate Change Mitigation (ETC/ACM) (Denby et al., 2005; Horálek et al., 2005, 2007, 2008, 2010; De Smet et al., 2010; Denby et al., 2011a,b). However, these maps are not routinely produced for NO_2 and only partially for O_3 and further do not provide any information on the temporal variability that can be found at a particular location throughout the year.

For this reason a prototype system for providing the approximate spatial and temporal patterns of background concentrations of PM_{10}, $PM_{2.5}$, O_3, and NO_2 over Norway has been developed at the Norwegian Institute for Air Research (NILU) in recent years (Schneider et al., 2011; Schneider and Obracaj, 2013; Schneider, 2013). The following sections summarize the basic principles of the system and describe some of the more recent work.

1.1 Basic outline of the system

The system for mapping background concentrations in Norway is based on two independent components, namely a spatial and a temporal component. Coupled together, these two components are intended to represent a typical year in Norway. The typical conditions are based on the idea of long-term averages in order to eliminate inter-annual variability. The spatial component consists of interpolated observations of background stations throughout Norway. For particulate matter gridded annual mean concentrations provided operationally by the EEA are used, whereas for the other two species (NO_2 and O_3) a geostatistical approach using several auxiliary datasets is applied in order to obtain the best possible estimates.

The temporal component is constructed using the average of the long-term time series of hourly observations at all relevant Norwegian stations for the various species. The time series generally have a length of between approximately five and fifteen years depending on species and the particular station. These datasets are acquired from the quality-controlled European air quality database *Airbase*.

A coupling of the two components was then carried out by averaging several years of hourly measurements on an annual- as well as on a daily basis. The resulting time series for a typical year and a typical day were further smoothed using a two-dimensional low-pass filter to ensure that the observations are representative of cyclical temporal patterns and do not just reflect short-term variability or outliers that are only present in a single year but do not reflect a typical situation. The representative annual and daily time series are subsequently converted from absolute concentrations given in μg m^{-3} to anomalies from the long-term mean at the station given in percent. This ensures the applicability of the temporal information for neighboring areas with differing mean annual background concentrations.

Due to the often short time series available at each station and the associated small sample size, random noise which is not representative of the overall long-term temporal variability is abundant in the time series and needs to be removed before using the relative anomalies for estimating concentrations at other locations. Such a task can for example be performed by using a moving average filter. However, for practical purposes this smoothing was performed here in the operational application by applying an asymmetric two-dimensional low-pass filter on an hour-by-hour anomaly matrix for an average year. This results in a simultaneous smoothing of both the annual and daily average time series. An example is shown in Figure 1. It should be noted that the application of the filter was performed while the matrix was augmented by itself on all four sides in order to avoid erroneous edge effects caused by the filter.

The smoothed relative anomalies can then be applied to neighboring locations with different absolute annual mean concentrations, and as such the average concentration can be estimated for a certain location given a certain day of the year and a time of day. The final report submitted to Klif/Miljødirektoratet for the 2011 work (Schneider et al., 2011) as well as follow-up reports (Schneider and Obracaj, 2013; Schneider, 2013) describe in detail the basic methodology of the system and some of the initial results. Figure 2 shows an example of the currently available online application of the background concentration mapping system.

1.2 Work carried out in 2014

The main goal of the work carried out in 2014 was centered around the automatization of the annual updating process in order to be able to accomplish this task relatively quickly (on the order of a few days per annual update cycle) and yet keep the consistent methodology developed previously. The updating of both the spatial and the temporal component of the system can be automated to a relatively large extent (the latter more than the former). While a certain manual effort is still required to acquire and prepare the necessary input data, extensive efforts have been made to automate the remaining processing steps to as large an extent as possible. These efforts are described in the following sections.

2 Practical aspects of the updating procedure

In practical terms, carrying out an annual update of the system is very straightforward and can easily be accomplished in less than two work days. Initially, the newest available data has to be acquired from the various data sources (primarily from the EEA but also from EMEP). Subsequently, the user must simply edit the required parameters in the Matlab script `BackgroundAtlas_Main.m` and run it. This main control script then performs the required preprocessing and calls the various subscripts in the required order.

Figure 3 shows how the various scripts are called by the main control script. All relevant code is further shown in Appendix A.

3 Automatization of the spatial component

While the annual updating of the temporal component could be fully automated, the spatial component was semi-automated due to the fact that the geostatistical

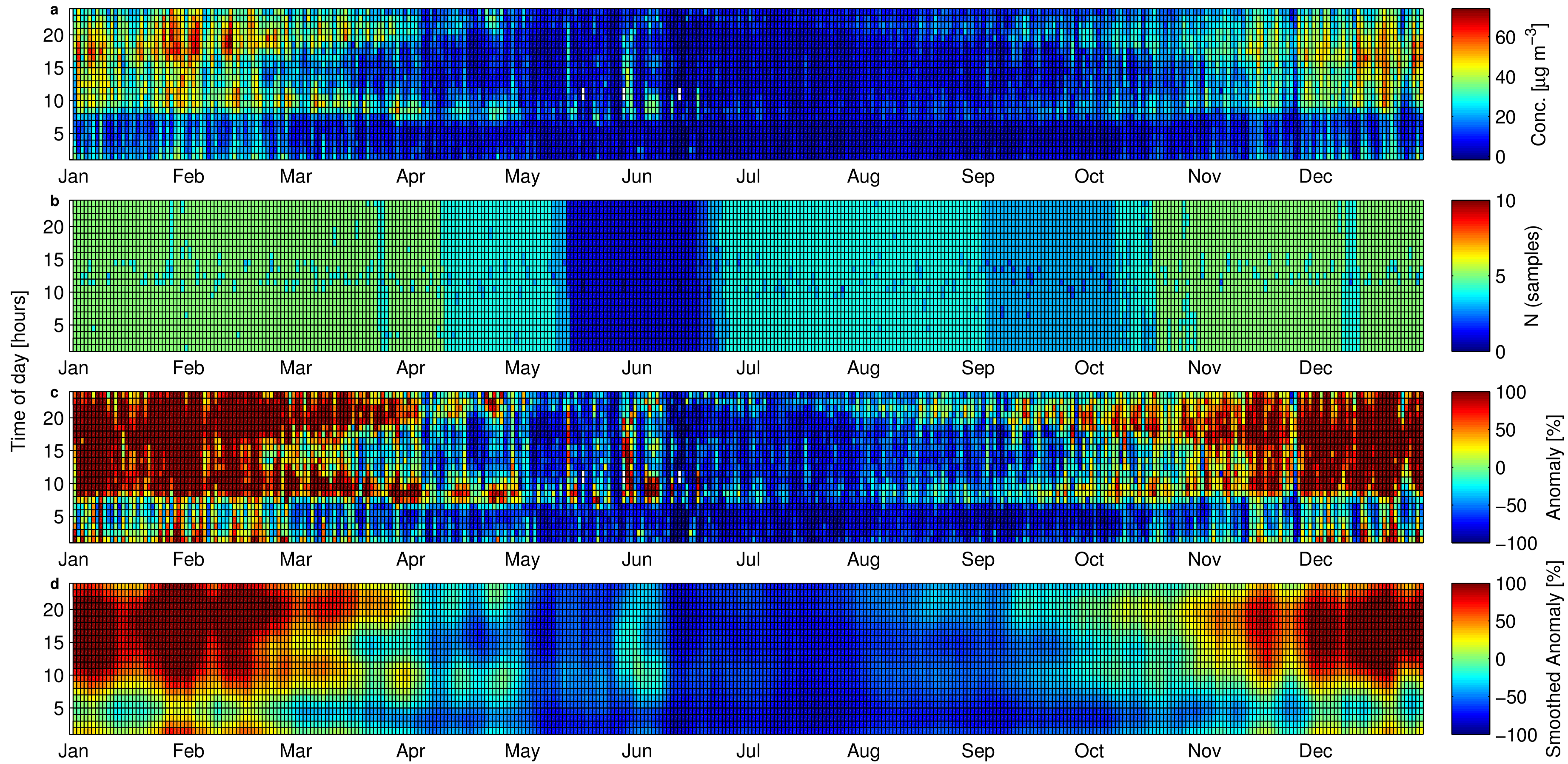

Figure 1 – NO_2 at station *NO0075A Barnehagen*: Annual matrices of hourly averages computed over entire available time series, shown as a) Observations, b) number of years with available data, c) the anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

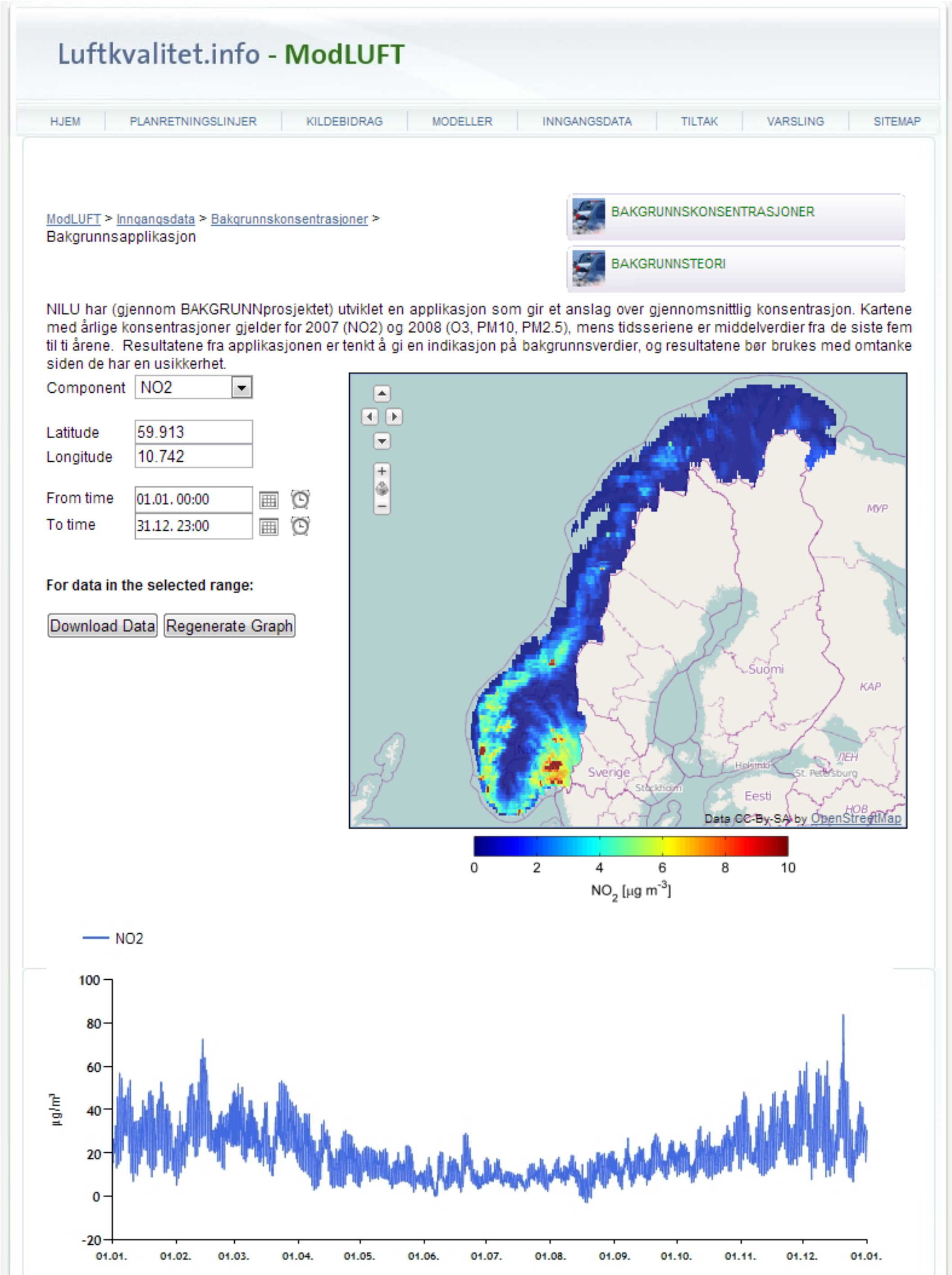

Figure 2 – Screenshot of the mapping component of the online web mapping application used for visualizing the results and providing access to the data, here showing background concentrations of NO_2 throughout all of Norway and the corresponding time series for central Oslo. The website can be found at `http://www.luftkvalitet.info/ModLUFT/Inngangsdata/Bakgrunnskonsentrasjoner/BAKGRUNNproj.aspx`.

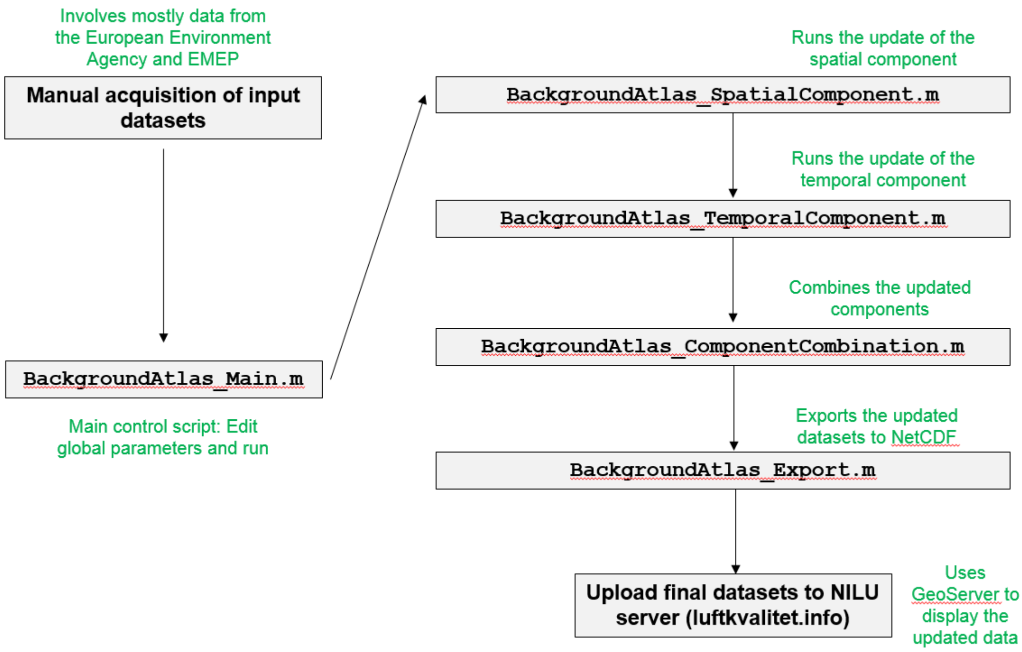

Figure 3 – Technical overview of the automated updating procedure for performing the annual updates of the background atlas.

approach for developing the annual average concentration maps in the spatial component requires some expert guidance to get the optimal results. However, besides some preparation of the input datasets and some visual interpretation of the semivariograms used for residual kriging, no further user interaction is necessary. Assuming no change in the format or content of the required input datasets (primarily those acquired from the European Environment Agency and the EMEP model), the spatial component can be updated in future with new annual data in one to two days.

3.1 Acquisition of input data

The system is based on a large variety of input datasets. An annual update requires that the most up-to-date version of these datasets are acquired by the user. In the following section the various input datasets are described and explanations about the corresponding updating procedures are given.

Figure 4 shows a directory tree of the system and describes the directory structure of the input datasets required for both the spatial and temporal components. Note that some non-essential subdirectories are not listed there for clarity. As can be seen, the spatial component requires several types of input datasets, including primarily GIS data from the EEA, modelling output from EMEP, annual average observations for all European stations and hourly raw observations for stations in Norway. Note that the scripts rely on the input data being stored exactly in the directory structure shown. However, the root folder of the system is flexible and can be specified in the script `BackgroundAtlas_Main.m`.

```
/rootDirectory
 |__ InputData                            This subdirectory contains all
 |    |                                   input data required for the
 |    |                                   entire system (both spatial and
 |    |                                   temporal component)
 |    |
 |    |__ SpatialComponent                Includes all datasets required
 |    |    |                              for running the spatial
 |    |    |                              component of the system
 |    |    |
 |    |    |__ Airbase                    Contains the annual statistics
 |    |    |                              of all Airbase stations in
 |    |    |                              Europe
 |    |    |
 |    |    |__ EEA_GIS_Data               Contains the annual average
 |    |    |                              concentration grids produced
 |    |    |                              by the ETC/ACM
 |    |    |
 |    |    |__ EMEP                       Contains the annual average
 |    |                                   concentration grids produced
 |    |                                   by the EMEP model
 |    |
 |    |__ TemporalComponent               Includes all datasets required
 |         |                              for running the temporal
 |         |                              component of the system
 |         |
 |         |__ Airbase
 |              |__ Norway                Contains the entire time series
 |                                        of hourly raw Airbase data for
 |                                        all stations in Norway
 |
 |__ Output                               All output files will be written
                                          into this subdirectory
```

Figure 4 – Directory tree of the system, mostly describing the various directories in which the relevant input data are stored. Note that some unessential subdirectories are not listed here.

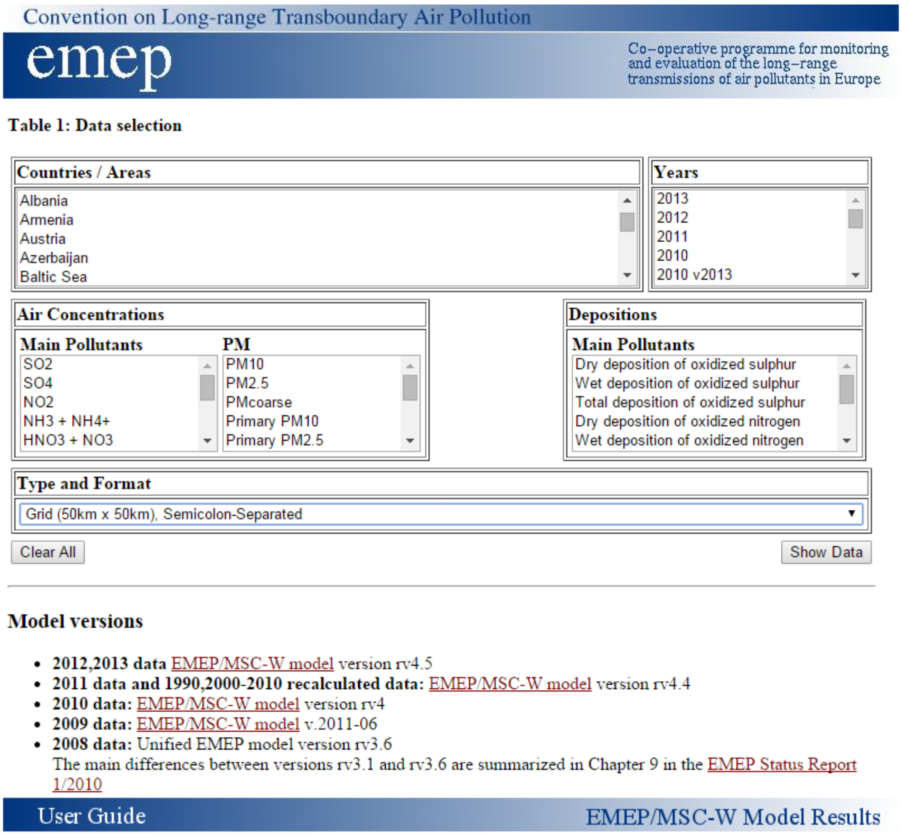

Figure 5 – Screenshot of the download interface for obtaining the annual average concentration fields for the EMEP model. The gridded CSV format has to be chosen.

3.1.1 Airbase GIS Data

Annual average GIS data grids for PM_{10} and $PM_{2.5}$ are acquired from the European Environment Agency (EEA). These maps are provided with a roughly 3-year time delay which is why this year (2014) data up to and including the year 2011 could be used.

The GIS data can be downloaded from `http://www.eea.europa.eu/data-and-maps/data/interpolated-air-quality-data-2`. The Shapefile version of the datasets named as `gis_data_2011_shapefile.zip` and similar for other years are needed. When extracted, these files contain the subdirectories `\data\20xx\pm10\` and `\data\20xx\pm25\` for which annual average GeoTIFF files for PM_{10} and $PM_{2.5}$ for the various years can be found. The GeoTIFFs then need to be converted from their projection to geographic coordinates with WGS84 datum before they can be stored in the input data directory (`\InputData\SpatialComponent\EEA_GIS_Data`) and be used by the Matlab script for the processing of the spatial component.

3.1.2 EMEP Modelling Results

Data supplied by the Unified EMEP model (Simpson et al., 2003) are essential for producing realistic concentration fields for those species that are not provided as part of the annual European air quality maps provided by the EEA.

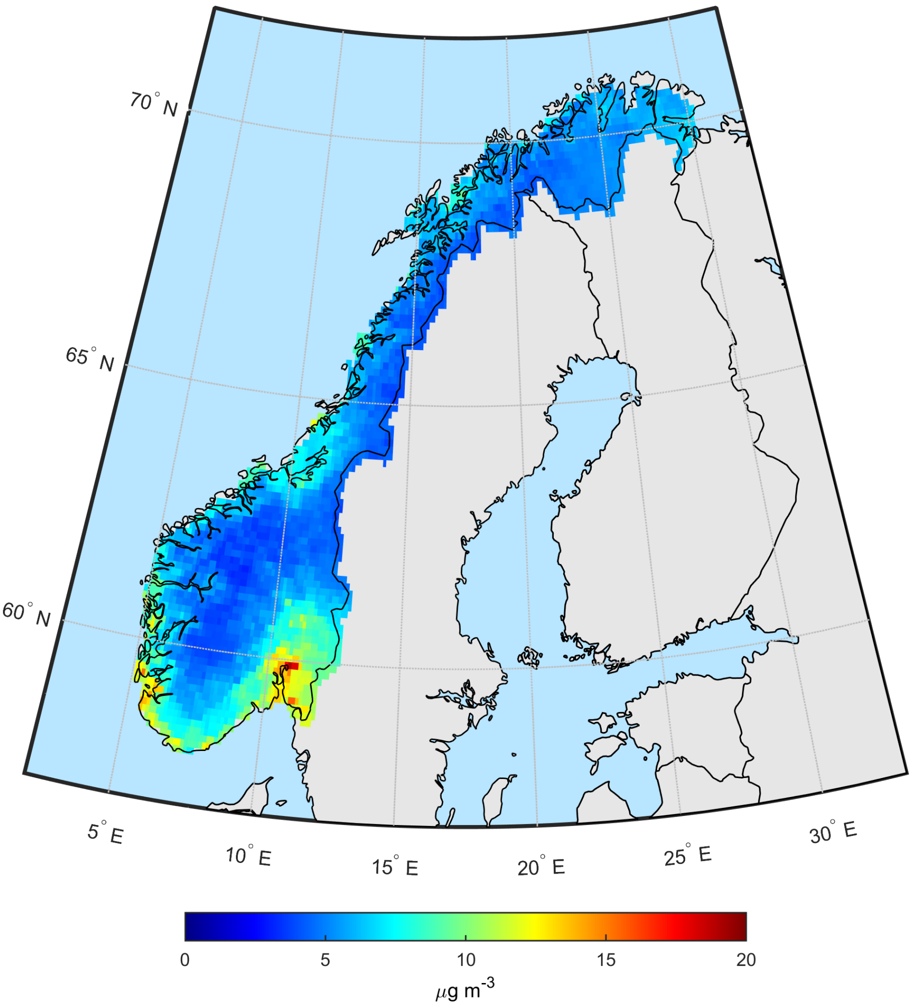

Figure 6 – Updated spatial patterns of PM_{10} concentrations in Norway as provided by the spatial component of the system. Here the average for the 3-year period 2009 through 2011 is shown.

The Unified EMEP model is a Eulerian chemical transport model that has been developed at the EMEP/MSC-W (Meteorological Synthesizing Centre West of EMEP) and has been extensively validated (Fagerli et al., 2003). Emissions used for the model are described in Vestreng et al. (2007). The modeled annual average concentrations were acquired as a grid with a 50 × 50 km horizontal spatial resolution. They were resampled to the final grid resolution of 10 km × 10 km used here through cubic convolution.

The species which need to be mapped as part of the system and which were not available from the EEA datasets are NO_2 and O_3. Annual average EMEP model concentrations for these two species are obtained in CSV format from `http://webdab.emep.int/Unified_Model_Results/`. Figure 5 shows a screenshot of the download interface in which the latest year has to be chosen and the annual average concentration fields for all four species have to be downloaded. These are then stored in the appropriate subdirectories in `\InputData\SpatialComponent\EMEP`. The import routine `read_emep.m` can then be used to read these datasets. This routine is automatically called by the primary processing script for updating the spatial component (`BackgroundAtlas_SpatialComponent.m`, see Appendix).

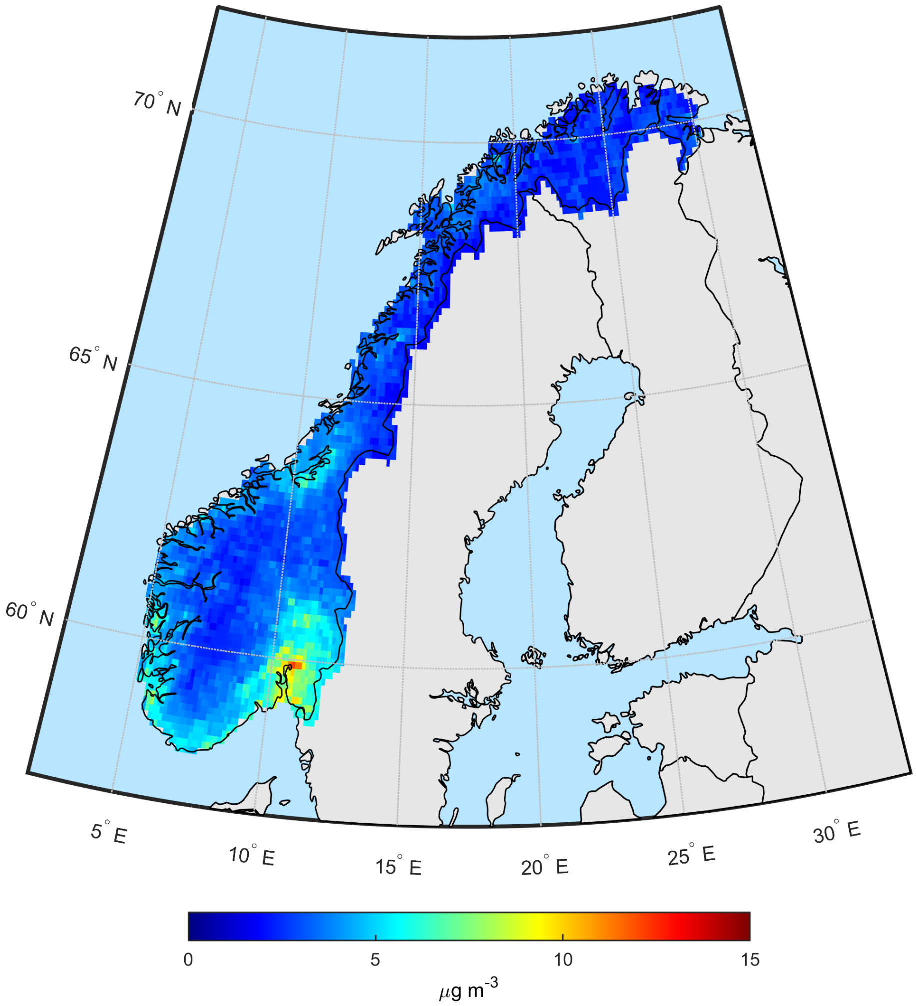

Figure 7 – Updated spatial patterns of $PM_{2.5}$ concentrations in Norway as provided by the spatial component of the system. Here the average for the 3-year period 2009 through 2011 is shown. Note that the color scale here has been slightly modified compared to PM_{10} in order to bring out more spatial detail.

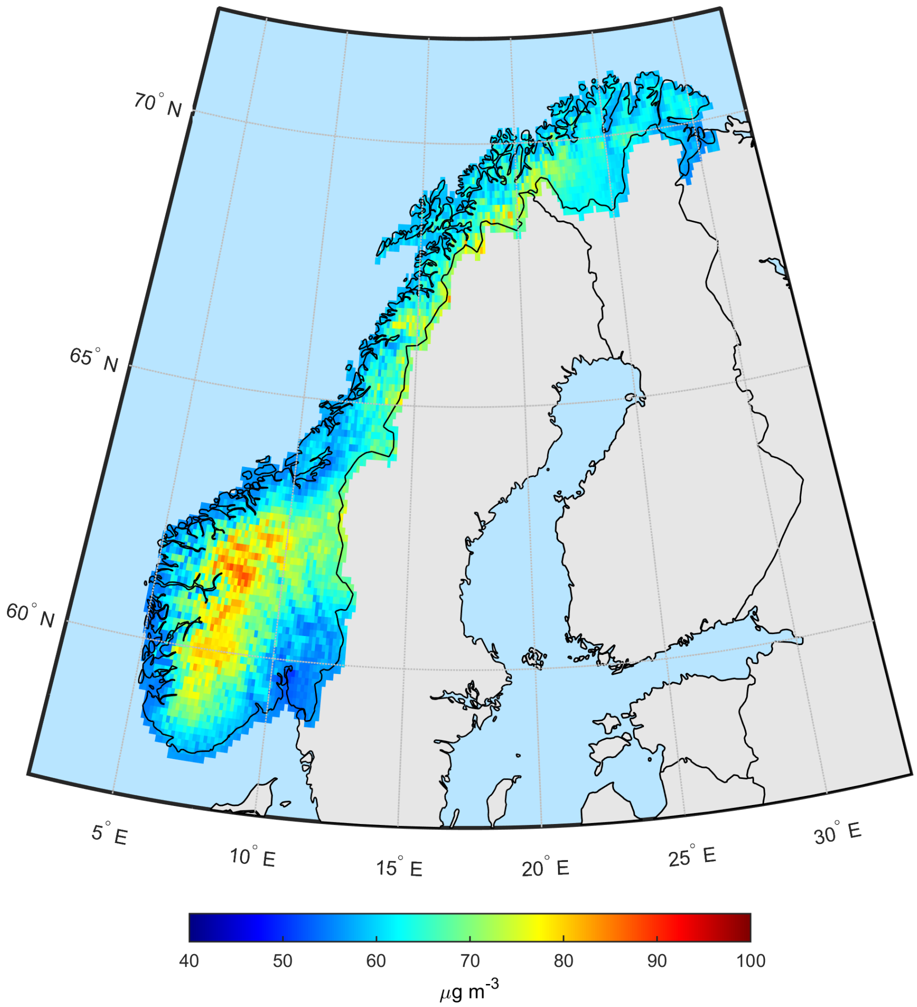

Figure 8 – Updated spatial patterns of O_3 concentrations in Norway as provided by the spatial component of the system. Here the average for the 3-year period 2009 through 2011 is shown.

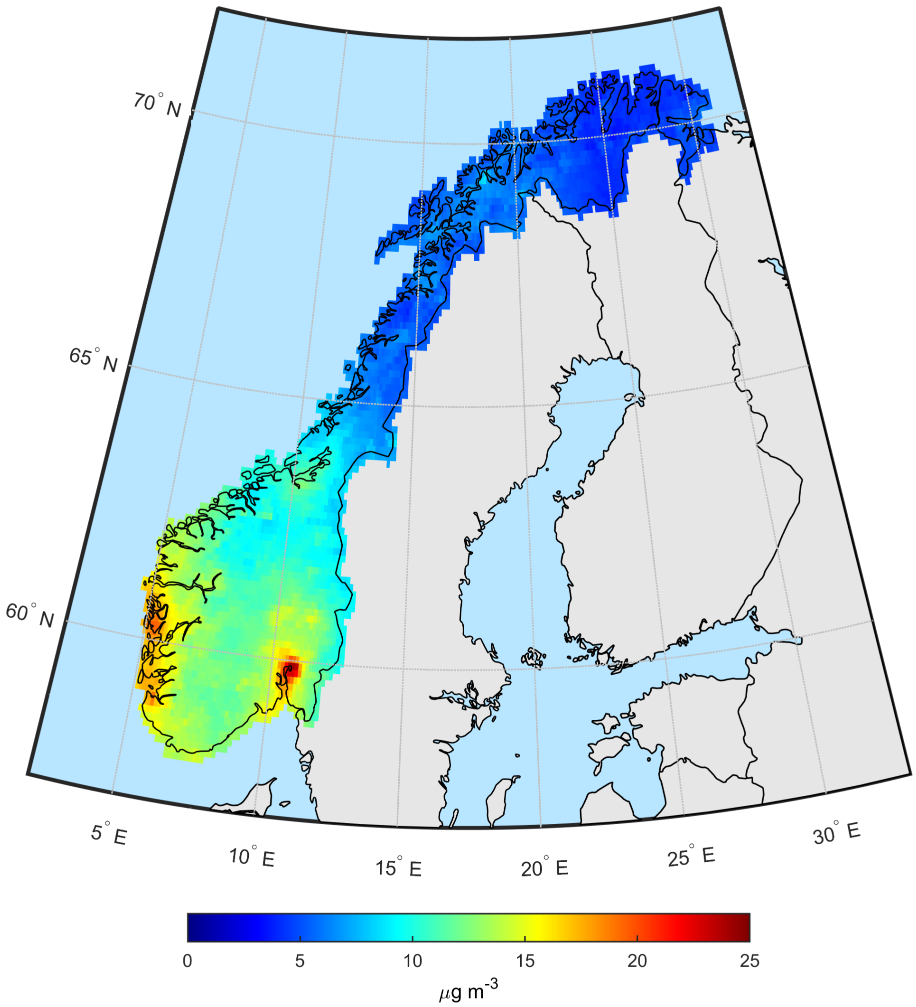

Figure 9 – Updated spatial patterns of NO_2 concentrations in Norway as provided by the spatial component of the system. Here the average for the 3-year period 2009 through 2011 is shown.

3.2 Updating the spatial component

The script BackgroundAtlas_SpatialComponent.m handles the entire update of the spatial component. Once the required input datasets have been acquired and are stored in the right subdirectories, this script will call all the necessary functions to perform the import, processing, and analysis. Note that this script does not need to be called by the user but is rather called by the main control script BackgroundAtlas_Main.m. Figures 6 through 9 show the updated 2009-2011 average concentrations for PM_{10}, $PM_{2.5}$, O_3, and NO_2 in Norway.

4 Automatization of the temporal component

The annual updating procedure for the temporal component could be automated to a very large extent. Only a single input dataset (country-level Airbase annual data file including hourly raw data) needs to be acquired by the user. No further manual processing is necessary. The main Matlab script for updating the temporal component (BackgroundAtlas_TemporalComponent.m) then updates the temporal component fully automatically based on this input dataset without any further user involvement.

4.1 Acquisition of input data

Raw data from air quality stations was used for both spatial mapping using residual kriging as well as for temporal decomposition of the time series. All station data was obtained from the *European Air quality dataBase*, AirBase (http://acm.eionet.europa.eu/databases/airbase/). However, different datasets were acquired for the spatial- versus the temporal component. For the geostatistical analysis and the mapping, annual mean concentrations were acquired for all European background stations in order to achieve a large enough sample size for variogram modeling and regression analysis.

For the temporal characterization, only data for Norwegian stations were acquired for all four species, however this was done for the entire available record and at an hourly temporal resolution. The only input data set required for the update of the temporal component is therefore the country-level raw Airbase dataset for Norway. This dataset differs from the AirBase dataset used in the spatial component as it does not provide annual average statistics at the European scale but rather provides the original raw data of hourly observations from all AirBase-supported stations in Norway.

This dataset can be found at the time of this writing at http://www.eea.europa.eu/data-and-maps/data/airbase-the-european-air-quality-database-8/. This dataset needs to be acquired and stored in the input data directory at \InputData\TemporalComponent\Airbase\Norway.

The Matlab scripts provided will automatically select the suitable background stations in Norway, import the data, and compute the average annual and daily anomalies at the hourly level.

4.2 Updating the temporal component

The temporal component was automated in such a way that only the latest version of the European Air Quality Database Airbase is necessary as an input. The Matlab script `BackgroundAtlas_TemporalComponent.m`, which is in turn called by `BackgroundAtlas_Main.m`, will automatically call required functions and routines to read, process, and analyse the temporal component. See also Figure 3 for an overview about how the various scripts interact with each other. See Appendix A for the specific code listings.

For completeness, Appendix B includes all the new updated anomaly matrices for all background stations in Norway, now containing observations up to and including the year 2013. Compared to the anomaly matrices provided in Schneider et al. (2011), these updated matrices now include three more years of data and thus are able to show temporal behavior that is much more representative of the typical conditions in Norway. Due to the higher number of years these anomaly matrices further have a significantly reduced number of outliers and similar issues with non-representative short-scale temporal variability.

5 Integration of spatial and temporal component and export

After the spatial and temporal component are updated individually, they need to be combined. This is performed primarily through computing the respective area of influence for each station and to then calculate the typical annual behavior at each grid cell based on a) the average annual mean at that location (obtained from the spatial component) and b) the long-term average temporal anomaly of the hourly values observed at the closest background air quality monitoring station for the respective species (obtained from the temporal component). This functionality is handled by the script `BackgroundAtlas_ComponentCombination.m`.

Schneider et al. (2011) as well as follow-on reports (Schneider and Obracaj, 2013; Schneider, 2013) provide more detail on how the combination of the spatial and temporal components of the system are handled, both in terms of theoretical considerations as well as practical implementation.

6 Data export

Finally, the script `BackgroundAtlas_Export.m` writes out the final datasets resulting from the combination of the spatial and the temporal component into self-describing NetCDF files and further exports annaul average maps in GeoTiff format, which are used as the main layers in GeoServer. The NetCDF files contain the typical hourly time series for each grid cell in Norway and are used by the online application to display the time series for the selected location.

NetCDF files are a very convenient, self-describing format for storing gridded spatio-temporal data. Figure 10 shows an example illustrating which variables are stored in the `pm10.nc` file by the `BackgroundAtlas_Export.m` script as well as their respective dimensions.

```
Source:
           N:\Inby\Aktive-prosjekter\o114057-Bakgrunn4\
 Output\pm10.nc
Format:
           classic
Global Attributes:
           Description   = 'Estimated mean background
                            concentration of PM10 for
                            3902 locations over Norway
                            given for all 8760 hours in
                            a typical year'
           Creation_Date = '25-Nov-2014 09:38:49'
Dimensions:
           rows    = 3902
           columns = 8760
Variables:
    Latitude
           Size:       3902x1
           Dimensions: rows
           Datatype:   double
    Longitude
           Size:       3902x1
           Dimensions: rows
           Datatype:   double
    Annual_mean_PM10
           Size:       3902x1
           Dimensions: rows
           Datatype:   double
    Day_of_year
           Size:       8760x1
           Dimensions: columns
           Datatype:   double
    Hour_of_day
           Size:       8760x1
           Dimensions: columns
           Datatype:   double
    PM10
           Size:       3902x8760
           Dimensions: rows,columns
           Datatype:   double
    Uncertainty
           Size:       3902x1
           Dimensions: rows
           Datatype:   double
```

Figure 10 – Example output of the `pm10.nc` file illustrating the different variables stored by the `BackgroundAtlas_Export.m` script and their various dimensions.

7 Conclusions

An automatization of the Norwegian spatial and temporal mapping system for background concentrations of PM_{10}, $PM_{2.5}$, NO_2, and O_3 has been carried out. This automatization allows for a simplified annual update procedure in future years, which will then be possible to be carried out in a short period on the order of 2 days. The user only needs to provide the input data sets (primarily EMEP model output and European-scale annual average air quality maps for PM_{10} and $PM_{2.5}$ provided by the European Environment Agency). The rest of the updating procedure is handled by Matlab scripts, which are to some extent provided in Appendix A.

While the updating procedure could be automated to a very large extent, a relatively small amount of manual user intervention is still necessary to a) acquire the most recent input datasets, b) ensure their validity and internal consistency, and c) to provide some expert guidance for the derivation of the theoretical semi-variogram used in the geostatistical mapping routines in the spatial component. Nonetheless the implemented automating procedures ensure that future updates of the system for Norwegian background concentrations can be carried out in a time-efficient manner.

While the current version of the background atlas can be updated very easily and in a relatively short time period, the system is to some extent still mostly based on the initial design decisions made in the initial project in 2011. As such, there are several aspects that in hindsight have proven to be less than ideal and should be improved in future work. These include two main aspects, namely the used mapping grid and the way station representativity is handled.

The original mapping grid was based on unprojected geographic coordinates. While this had some initial practical advantages related to the combined handling of several datasets, in turned out that for geostatistical applications as they are required as part of the spatial component, unprojected datasets are less than ideal. Furthermore the unprojected grid resulted in rectangular grid cells of uneven size throughout Norway. It is therefore recommended to replace the existing unprojected grid by a projected grid in the Universal Transverse Mercator (UTM) projection with WGS84 datum.

In addition, the original system simply used the nearest station to determine the temporal behavior at any given point in Norway. While this is a reasonable first approach, it is recommended to investigate possible methods for improving this technique, for example by computing a linear combination of the typical time series from several surrounding stations.

In combination these two changes have the potential to drastically improve the usefulness of the system for providing background concentrations in Norway.

References

de Smet, P., Horálek, J., Conková, M., Kurfürst, P., de Leeuw, F, Denby, B. (2010) European air quality maps of ozone and PM10 for 2008 and their uncertainty analysis. Bilthoven, European Topic Centre on Air and Climate Change (ETC/ACC Technical paper, 2010/10).

Denby, B., Gola, G., de Leeuw, F., de Smet, P., Horálek, J. (2011) Calculation of pseudo PM2.5 annual mean concentrations in Europe based on annual mean PM10 concentrations and other supplementary data. Bilthoven, European Topic Centre on Air and Climate Change (ETC/ACC Technical Paper, 2010/9).

Denby, B., Horálek, J., Walker, S.E., Eben, K., Fiala, J. (2005) Interpolation and assimilation methods for European scale air quality assessment and mapping. Part 1: Review and recommendations. Final draft. Bilthoven, European Topic Centre on Air and Climate Change (ETC/ACC Technical Paper, 2005/7).

Denby, B., Horálek, J., de Smet, P., de Leeuw, F. (2011) Mapping annual mean PM2.5 concentrations in Europe: application of pseudo PM2.5 station data. Bilthoven, European Topic Centre on Air and Climate Change (ETC/ACM Technical Paper, 2011/5).

Fagerli, F., Simpson, D., Tsyro, S., Solberg, S., Aas, W. (2003) Transboundary acidification, eutrophication and ground level ozone in Europe. Part II. Unified EMEP model performance. EMEP status report 1/2003. Oslo, Norwegian Meteorological Institute (EMEP report, 1/2003).

Horálek, J., de Smet, P., de Leeuw, F., Conková, M., Denby, B., Kurfürst, P. (2010) Methodological improvements on interpolating European air quality maps. Bilthoven, European Topic Centre on Air and Climate Change (ETC/ACC Technical Paper, 2009/16).

Horálek, J., de Smet, P., de Leeuw, F., Denby, B., Kurfürst, P., Swart, R. (2008) European air quality maps for 2005 including uncertainty analysis. Bilthoven, European Topic Centre on Air and Climate Change (ETC/ACC Technical Paper, 2007/7).

Horálek, J., Denby, B., de Smet, P., de Leeuw, F., Kurfürst, P., Swart, R., van Noije, T. (2007) Spatial mapping of air quality for European scale assessment. Bilthoven, European Topic Centre on Air and Climate Change (ETC/ACC Technical Paper 2006/6).

Horálek, J., Kurfürst, P., Denby, B., de Smet, P., de Leeuw, F., Brabec, M., Fiala, J. (2005) Interpolation and assimilation methods for European scale air quality assessment and mapping. Part II: Development and testing new methodologies. Final draft. Bilthoven, European Topic Centre on Air and Climate Change (ETC/ACC Technical Paper, 2005/8).

Schneider, P. (2013) Background concentrations in Norway - Temporal averaging and uncertainty assessment. Kjeller, NILU (NILU OR, 58/2013).

Schneider, P., Obracaj, A. (2013) Evaluation of new data sources for improving the estimation of background concentrations in Norway. Kjeller, NILU (NILU OR, 01/2013).

Schneider, P., Tønnesen, D., Denby, B. (2011) Update of background concentrations over Norway. Kjeller, NILU (NILU OR, 68/2011).

Simpson, D., Fagerli, H., Jonson, J., Tsyro, S., Wind, P. (2003). Transboundary acidification, eutrophication and ground level ozone in Europe.Part I. Unified EMEP model description. EMEP status report 1/2003. Oslo, Norwegian Meteorological Institute (EMEP report, 1/2003).

Vestreng, V., Mareckova, K., Kakareka, S., Malchykhina, A., Kukharchyk, T. (2007). Inventory review 2007. Emission data reported to LRTAP Convention and NEC Directive - Stage 1 and 2 review. Oslo, Norwegian Meteorological Institute - Meteorological Synthesizing Centre-West, MSC-W (Technical Report, 1/2007).

Acknowledgments

Funding provided under the Miljødirektoratet contract 14078154 is gratefully acknowledged.

Appendices

A Code listings

The following section provides code snippets and functions written in the Matlab programming language that were developed for achieving a semi-automatization of the updating procedure of the Norwegian atlas of background conncentrations.

A.1 Main control script

Listing 1 – Main control script: `BackgroundAtlas_Main.m`

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%% Norwegian Background Cocentrations %%%%%%%%%%
%%%%%%%%%%%%%%%% Main Control Script %%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% User-defined parameters

%% GENERAL
% Define the root folder containing the main folder structure
rootfolder = 'N:\Inby\Aktive-prosjekter\o114057-Bakgrunn4\';

%% SPATIAL COMPONENT
% Define averaging period for spatial component
begYear = 2009; % First year of averaging period
endYear = 2011; % Last year of averaging period

%% TEMPORAL COMPONENT
min_number_of_years = 3; % Minimum number of years with available
    data before a station is considered suitable for the temporal
    component


%% RUN THE SPATIAL COMPONENT
run('BackgroundAtlas_SpatialComponent')

%% RUN THE TEMPORAL COMPONENT
run('BackgroundAtlas_TemporalComponent')

%% COMBINE THE OUTPUT FROM SPATIAL AND TEMPORAL COMPONENT
run('BackgroundAtlas_ComponentCombination')

%% OUTPUT THE RESULTS
run('BackgroundAtlas_Export')

disp('Update completed.')
```

A.2 Spatial Component

Listing 2 – Main script for the spatial component: `BackgroundAtlas_SpatialComponent.m`

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%% SPATIAL COMPONENT %%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Todo: Ideally upgrade the kriging to use UTM coordinates?

%% Initialization
cd(rootfolder);


% Initalize latitude longitude grids
% Get the lat/lon grids from the PM2.5 grid
bbox = [20 73 -20 40];
[pm25_2008 R] = geotiffread([rootfolder 'InputData\SpatialComponent\
    EEA_GIS_Data\PM25\EEA_10kmgrid_2008_pm25_avg_wgs84.tif']);
[lat lon] = meshgrat(pm25_2008, R);
[mask R] = geotiffread([rootfolder 'InputData\SpatialComponent\
    Auxiliary\norway_mask.tif']);
mask(mask==0) = NaN;
lat_no = lat(1:120,240:500);
lon_no = lon(1:120,240:500);
mask_no = mask(1:120,240:500);


%% Read static datasets
% DEM
load([rootfolder 'InputData\SpatialComponent\Auxiliary\
    digital_elevation_model.mat']);


% Compute 3 year average for PM10
years = [begYear:endYear];
for i = 1:length(years)
    year = years(i);
    geotifffile = [rootfolder 'InputData\SpatialComponent\
        EEA_GIS_Data\PM10\EEA_10kmgrid_' num2str(year) '
        _pm10_avg_wgs84.tif'];
    [pm10_stack(:,:,i) R] = geotiffread(geotifffile);
end
pm10_stack(pm10_stack == 0) = NaN;
[emeplat emeplon] = meshgrat(pm10_stack(:,:,1), R);
pm10_avg_temp = nanmean(pm10_stack,3);
pm10_avg = interp2(emeplon, emeplat, pm10_avg_temp, lon_no, lat_no,
    'cublic');


% Compute 3 year average for PM25
years = [begYear:endYear];
for i = 1:length(years)
    year = years(i);
    geotifffile = [rootfolder 'InputData\SpatialComponent\
        EEA_GIS_Data\PM25\EEA_10kmgrid_' num2str(year) '
        _pm25_avg_wgs84.tif'];
    [pm25_stack(:,:,i) R] = geotiffread(geotifffile);
end
pm25_stack(pm25_stack == 0) = NaN;
[emeplat emeplon] = meshgrat(pm25_stack(:,:,1), R);
pm25_avg_temp = nanmean(pm25_stack,3);
pm25_avg = interp2(emeplon, emeplat, pm25_avg_temp, lon_no, lat_no,
    'cublic');



% Generate grids of Ozone concentrations

% Read entire Airbase data base statistics (not hourly data)
```

```
meta_europe = readAirbaseStationFile([rootfolder 'InputData\
    SpatialComponent\Airbase\Europe\AirBase_v8_stations.csv']);

years = [begYear:endYear];
for j = 1:length(years)
    year = years(j);

    % Get Airbase data for the current year
    [airbase airbase_bg] = readAirbaseStatsFile([rootfolder '
        InputData\SpatialComponent\Airbase\Europe\
        AirBase_v8_statistics.csv'], meta_europe, year);

    o3 = [airbase_bg.o3]';
    lat = [airbase_bg.station_latitude_deg]';
    lon = [airbase_bg.station_longitude_deg]';
    alt = [airbase_bg.station_altitude]';
    pos = [lon lat];

    % Read EMEP data
    emepfilename = [rootfolder 'InputData\SpatialComponent\EMEP\O3\
        EMEP_O3_' num2str(year) '.txt'];
    [emeplat emeplon emepo3] = readEMEP(emepfilename);
    bbox = [20 73 -20 40];
    resolution = 0.25;
    [xi,yi] = meshgrid(bbox(3):resolution:bbox(4),bbox(1):resolution
        :bbox(2));
    xic = reshape(xi, numel(xi), 1);
    yic = reshape(yi, numel(yi), 1);
    pos_est = [xic yic];
    emepi = griddata(emeplon, emeplat, emepo3, xi, yi, 'cubic');

    % extract emep O3 at stations
    for i=1:length(airbase_bg)
        [row col minval] = ll2rowcol(yi, xi, airbase_bg(i).
            station_latitude_deg, airbase_bg(i).
            station_longitude_deg);
        if row <= size(emepi,1) & col <= size(emepi,2)
            emepo3_at_station(i,1) = emepi(row,col);
        else
            emepo3_at_station(i,1) = NaN;
        end
        %disp(i)
    end

    % Rcompute emepi just for Norway
    emepi = griddata(emeplon, emeplat, emepo3, lon_no, lat_no, '
        cubic');

    rst = regstats(o3, [alt emepo3_at_station], 'linear');
    regr_o3 = rst.beta(1) + rst.beta(2) .* demi + rst.beta(3) .*
        emepi;

    % Fit semivariogram
    [S V] = semiAutoVario(pos, rst.r);

    % krige residuals
    good = ~isnan(rst.r);
    xic = reshape(lon_no, numel(lon_no), 1);
    yic = reshape(lat_no, numel(lat_no), 1);
    pos_est = [xic yic];
    [r_est,r_var]=gstat_krig(pos(good,:),rst.r(good),pos_est,V);
    rout = reshape(r_est, size(lat_no));
    rout_var = reshape(r_var, size(lat_no));
```

```
    temp_final_o3 = rout + regr_o3;
    grid_o3(:,:,j) = temp_final_o3 .* mask_no;
    grid_se_o3 = sqrt(rout_var) .* mask_no;
    masko3 = double(~isnan(grid_o3(:,:,j)));
    masko3(masko3==0) = NaN;
    grid_se_o3(:,:,j) = grid_se_o3 .* masko3;
    disp(['O3:␣Year␣' num2str(year) '␣finished.'])
end
o3_avg = nanmean(grid_o3,3);



% Generate grids of NO2 concentrations

years = [begYear:endYear];
for j = 1:length(years)
    year = years(j);

    %Get Airbase data for the current year
    [airbase airbase_bg] = readAirbaseStatsFile([rootfolder '
       InputData\SpatialComponent\Airbase\Europe\
       AirBase_v8_statistics.csv'], meta_europe, year);

    no2 = [airbase_bg.no2]';
    lat = [airbase_bg.station_latitude_deg]';
    lon = [airbase_bg.station_longitude_deg]';
    alt = [airbase_bg.station_altitude]';
    pos = [lon lat];

    %Read EMEP data
    emepfilename = [rootfolder 'InputData\SpatialComponent\EMEP\NO2\
       EMEP_NO2_' num2str(year) '.txt'];
    [emeplat emeplon emepno2] = readEMEP(emepfilename);
    bbox = [20 73 -20 40]
    resolution = 0.25;
    [xi,yi] = meshgrid(bbox(3):resolution:bbox(4),bbox(1):resolution
       :bbox(2));
    xic = reshape(xi, numel(xi), 1);
    yic = reshape(yi, numel(yi), 1);
    pos_est = [xic yic];
    emepi = griddata(emeplon, emeplat, emepno2, xi, yi, 'cubic');

    %extract emep NO2 at stations
    for i=1:length(airbase_bg)
        [row col minval] = ll2rowcol(yi, xi, airbase_bg(i).
           station_latitude_deg, airbase_bg(i).
           station_longitude_deg);
        if row <= size(emepi,1) & col <= size(emepi,2)
            emepno2_at_station(i,1) = emepi(row,col);
        else
            emepno2_at_station(i,1) = NaN;
        end
        disp(i)
    end

    % Read population density for Europe
    pdfile = 'H:\DataArchive\GPWv3\euds00ag.asc';
    [pd R] = arcgridread(pdfile);
    [pdlon pdlat] = pixcenters(R, size(pd), 'makegrid');
    pd(isnan(pd)) = 0;
    pdi = griddata(pdlon, pdlat, pd, xi, yi, 'cubic');
```

```
    for i=1:length(airbase_bg)
        [row col minval] = ll2rowcol(pdlat, pdlon, airbase_bg(i).
            station_latitude_deg, airbase_bg(i).
            station_longitude_deg);
        [row col] = getrowcol(airbase_bg(i).station_latitude_deg,
            airbase_bg(i).station_longitude_deg, pdlat, pdlon, pd);
        airbase_v7_2008_bg(i).pd = pd(row,col);
        pd_at_station(i,1) = pd(row,col);
        disp(i)
    end

    %OMI NO2
    load OMI_NASA_highres_Annual_Mean_2009_from_daily.mat
    %lathr = repmat(lathr, 1, size(lonhr,1));
    %lonhr = repmat(lonhr, 1, size(lathr,1))';
    for i=1:length(airbase_bg)
        [row col] = getrowcol(airbase_bg(i).station_latitude_deg,
            airbase_bg(i).station_longitude_deg, lathr, lonhr,
            omino2e2009hr);
        omino2_atstation(i,1) = omino2e2009hr(row,col);
        disp(i)
    end

    emepi_no = griddata(emeplon, emeplat, emepno2, lon_no, lat_no, '
        cubic');
    pdi_no = griddata(pdlon, pdlat, pd, lon_no, lat_no, 'cubic');
    omii_no = griddata(lonhr,lathr,omino2e2009hr,lon_no,lat_no,'
        cubic');

    rst = regstats(no2, [omino2_atstation emepno2_at_station
        pd_at_station], [1 0 0; 0 1 0; 0 0 1]);
    regr_no2 = rst.beta(1) .* omii_no + rst.beta(2) .* emepi_no +
        rst.beta(3) .* pdi_no;

    %Fit semivariogram
    [S V] = semiAutoVario(pos, rst.r);

    %krige residuals
    good = ~isnan(rst.r);
    xic = reshape(lon_no, numel(lon_no), 1);
    yic = reshape(lat_no, numel(lat_no), 1);
    pos_est = [xic yic];
    [r_est,r_var]=gstat_krig(pos(good,:),rst.r(good),pos_est,V);
    rout = reshape(r_est, size(lat_no));
    rout_var = reshape(r_var, size(lat_no));

    temp_final_no2 = rout + regr_no2;
    grid_no2(:,:,j) = temp_final_no2 .* mask_no;
    grid_se_no2 = sqrt(rout_var) .* mask_no;
    maskno2 = double(~isnan(grid_no2(:,:,j)));
    maskno2(maskno2==0) = NaN;
    grid_se_no2(:,:,j) = grid_se_no2 .* maskno2;

    disp(['NO2:␣Year␣' num2str(year) '␣finished.'])
end
no2_avg = nanmean(grid_no2,3);



% Mask the average grids
no2_avg = no2_avg .* mask_no;
no2_avg(no2_avg < 0) = NaN;
```

```
o3_avg = o3_avg .* mask_no;
o3_avg(o3_avg < 0) = NaN;
pm25_avg = pm25_avg .* mask_no;
pm25_avg(pm25_avg < 0 ) = NaN;
pm10_avg = pm10_avg .* mask_no;
pm10_avg(pm10_avg < 0 ) = NaN;
```

Listing 3 – Reads in Airbase station metadata

```
function meta = readAirbaseStationFile(station_metadata_file)
% Reads in Airbase station metadata

fid = fopen(station_metadata_file)
    header = textscan(fid, '%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%
        s␣%s␣%s␣%s␣%s␣%s␣%s', 1, 'Delimiter', '\t', 'CollectOutput',
         1)
    metaraw = textscan(fid, '%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%s␣%f␣
        %f␣%f␣%s␣%s␣%s␣%s␣%s', 'Delimiter', '\t', 'HeaderLines', 1)
fclose(fid)
fieldnames = header{1};

% now create the metadata struct
nStations = length(metaraw{1});
nFields = length(fieldnames);
for i=1:nStations
    for j=1:nFields
        if ~isnumeric(metaraw{j}(i))
            meta(i,1).(fieldnames{j}) = metaraw{j}{i};
        else
            meta(i,1).(fieldnames{j}) = metaraw{j}(i);
        end
    end
end

%save 'C:\Users\ps\Dropbox\Work\NILU\Data\Matlab Data\
   Airbase_Metadata_v7.mat' meta


end
```

A.3 Temporal Component

Listing 4 – Main script for the temporal component: `BackgroundAtlas_TemporalComponent.m`

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%% TEMPORAL COMPONENT %%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%cd(airbase_rootfolder)


% Read in metadata for the Norwegian background stations
meta = readCSVinStruct([rootfolder 'InputData\TemporalComponent\
   Airbase\Norway\AirBase_NO_v8_stations.csv'], '\t');
ind = find(strcmp('NO', {meta.country_iso_code}') & strcmp('
   Background', {meta.type_of_station}'));

% Fix lat/lon of Stener Heyerdahl station
```

```
meta(find(strcmp({meta.station_european_code}, 'NO0063A'))).
    station_latitude_deg = 58.14888; % Coordinates measured in situ
    by CH
meta(find(strcmp({meta.station_european_code}, 'NO0063A'))).
    station_longitude_deg = 7.99183; % Coordinates measured in situ
    by CH

% Find all the files with raw data
%cd(airbase_rawdatafolder)
%filenames = dir(airbase_rawdatafolder);
%files = rdir(airbase_rawdatafolder);
%files = files(3:end);
airbase_rawdatafolder = [rootfolder 'InputData\TemporalComponent\
    Airbase\Norway\AirBase_NO_v8_rawdata\'];
filenames = dir(airbase_rawdatafolder);
filenames = {filenames(3:end).name}';
for i=1:length(filenames)
    station_european_code{i,1} = filenames{i}(1:7);
    component_code(i,1) = str2num(filenames{i}(8:12));
    datatype{i,1} = filenames{i}(18:22);
end
for i=1:length(filenames)
    metaind(i,1) = strmatch(station_european_code(i), {meta.
        station_european_code}');
    if strmatch(meta(metaind(i)).type_of_station, 'Background');
        isBackground(i,1) = 1;
    else
        isBackground(i,1) = 0;
    end
end

% Get stations with suitable record length for NO2
no2filesind = find(component_code == 8 & isBackground == 1 & strncmp
    ('hour.', datatype, 5));
for i=1:length(no2filesind)
    [sd data] = importHourlyRawAirbase([airbase_rawdatafolder
        filenames{no2filesind(i)}]);
    reclengthno2(i,1) = length(~isnan(data))/24/30/12;
    disp(['NO2,␣station␣number␣' num2str(i)]);
end
no2filesind = no2filesind(reclengthno2 > min_number_of_years); %
    stations with more than N years of data
no2stations = station_european_code(no2filesind);
no2meta = meta( metaind(no2filesind) );

o3filesind = find(component_code == 7 & isBackground == 1 & strncmp(
    'hour.', datatype, 5));
for i=1:length(o3filesind)
    [sd data] = importHourlyRawAirbase([airbase_rawdatafolder
        filenames{o3filesind(i)}]);
    reclengtho3(i,1) = length(~isnan(data))/24/30/12;
    disp(['O3,␣station␣number␣' num2str(i)]);
end
o3filesind = o3filesind(reclengtho3 > min_number_of_years);
o3stations = station_european_code(o3filesind);
o3meta = meta( metaind(o3filesind) );

pm10filesind = find(component_code == 5 & isBackground == 1 &
    strncmp('hour.', datatype, 5));
for i=1:length(pm10filesind)
    [sd data] = importHourlyRawAirbase([airbase_rawdatafolder
        filenames{pm10filesind(i)}]);
    reclengthpm10(i,1) = length(~isnan(data))/24/30/12;
```

```
    disp(['PM10,␣station␣number␣' num2str(i)]);
end
pm10filesind = pm10filesind(reclengthpm10 > min_number_of_years);
pm10stations = station_european_code(pm10filesind);
pm10meta = meta( metaind(pm10filesind) );

pm25filesind = find(component_code == 6001 & isBackground == 1 &
    strncmp('hour.', datatype, 5));
for i=1:length(pm25filesind)
    [sd data] = importHourlyRawAirbase([airbase_rawdatafolder
        filenames{pm25filesind(i)}]);
    reclengthpm25(i,1) = length(~isnan(data))/24/30/12;
    disp(['PM2.5,␣station␣number␣' num2str(i)]);
end
pm25filesind = pm25filesind(reclengthpm25 > min_number_of_years);
pm25stations = station_european_code(pm25filesind);
pm25meta = meta( metaind(pm25filesind) );


% Read in the data for the four species
no2 = no2meta;
for i=1:length(no2filesind)
    [no2(i).sd no2(i).data] = importHourlyRawAirbase([
        airbase_rawdatafolder filenames{no2filesind(i)}]);
    disp(['Reading␣NO2,␣station␣number␣' num2str(i)]);
end

o3 = o3meta;
for i=1:length(o3filesind)
    [o3(i).sd o3(i).data] = importHourlyRawAirbase([
        airbase_rawdatafolder filenames{o3filesind(i)}]);
    disp(['Reading␣O3,␣station␣number␣' num2str(i)]);
end

pm10 = pm10meta;
for i=1:length(pm10filesind)
    [pm10(i).sd pm10(i).data] = importHourlyRawAirbase([
        airbase_rawdatafolder filenames{pm10filesind(i)}]);
    disp(['Reading␣PM10,␣station␣number␣' num2str(i)]);
end

pm25 = pm25meta;
for i=1:length(pm25filesind)
    [pm25(i).sd pm25(i).data] = importHourlyRawAirbase([
        airbase_rawdatafolder filenames{pm25filesind(i)}]);
    disp(['Reading␣PM2.5,␣station␣number␣' num2str(i)]);
end


%% Filter out some stations with too big data gaps
for i=1:length(no2)
    [dailycycle dailycycleN yearlycycleday yearlycycledayN
        hourlymean hourlymeanN(:,:,i)] = decompose_timeseries(no2(i)
        .sd,  no2(i).data);
end
hourlymeanN(hourlymeanN < 3) = NaN;
nDaysWithNoData = squeeze(sum(sum(isnan(hourlymeanN),2) == 24));
no2 = no2(find(nDaysWithNoData < 60 )); % Take 2 months as threshold
     for now
clear nDaysWithNoData dailycycle dailycycleN yearlycycleday
    yearlycycledayN hourlymean hourlymeanN

for i=1:length(o3)
```

```
    %[dailycycle yearlycycleday hourlymean hourlymeanN(:,:,i)] =
        decompose_timeseries(o3(i).sd,  o3(i).data);
    [dailycycle dailycycleN yearlycycleday yearlycycledayN
        hourlymean hourlymeanN(:,:,i)] = decompose_timeseries(o3(i).
        sd,  o3(i).data);
    disp(i)
end
hourlymeanN(hourlymeanN < 3) = NaN;
nDaysWithNoData = squeeze(sum(sum(isnan(hourlymeanN),2) == 24));
o3 = o3(find(nDaysWithNoData < 60 )); % Take 2 months as threshold
    for now
clear nDaysWithNoData dailycycle dailycycleN yearlycycleday
    yearlycycledayN hourlymean hourlymeanN

for i=1:length(pm10)
    [dailycycle dailycycleN yearlycycleday yearlycycledayN
        hourlymean hourlymeanN(:,:,i)] = decompose_timeseries(pm10(i
        ).sd,  pm10(i).data);
end
hourlymeanN(hourlymeanN < 3) = NaN;
nDaysWithNoData = squeeze(sum(sum(isnan(hourlymeanN),2) == 24));
pm10 = pm10(find(nDaysWithNoData < 60 )); % Take 2 months as
    threshold for now
clear nDaysWithNoData dailycycle dailycycleN yearlycycleday
    yearlycycledayN hourlymean hourlymeanN

for i=1:length(pm25)
    [dailycycle dailycycleN yearlycycleday yearlycycledayN
        hourlymean hourlymeanN(:,:,i)] = decompose_timeseries(pm25(i
        ).sd,  pm25(i).data);
end
hourlymeanN(hourlymeanN < 3) = NaN;
nDaysWithNoData = squeeze(sum(sum(isnan(hourlymeanN),2) == 24));
pm25 = pm25(find(nDaysWithNoData < 60 )); % Take 2 months as
    threshold for now
clear nDaysWithNoData dailycycle dailycycleN yearlycycleday
    yearlycycledayN hourlymean hourlymeanN



% Generate smooth 3D matrices of anomalies for each species
for i=1:length(no2)
    m = nanmean(no2(i).data);
    [dailycycle dailycycleN yearlycycleday yearlycycledayN
        hourlymean hourlymeanN] = decompose_timeseries(no2(i).sd,
        no2(i).data);

    % Using hourlmean matrix and smoothing
    hourlymean = naninterp(hourlymean); % first fix NaNs
    hourlymean_anomaly = (hourlymean - m) ./ m * 100; % Compute
        anomaly

    % Duplicate the matrix by itself on all sides to avoid edge
        effects
    temp = repmat(hourlymean_anomaly,3,3);

    % Create the smoothing filter
    %h=ones(7)/49; % Create filter
    h = ones(3,7)./(3*7);

    % Filter the matrix
    smoothtemp= filter2(h, temp); % smooth the extended matrix
```

```
    % Extract original matrix from the smoothed results
    no2anomaly(:,:,i) = smoothtemp(366:366+364,25:25+23);
    no2anomalyN(:,:,i) = hourlymeanN;

    disp(['Generating␣anomaly␣matrix␣for␣NO2,␣station␣number␣'
       num2str(i)]);
end

for i=1:length(o3)
    m = nanmean(o3(i).data);
    [dailycycle dailycycleN yearlycycleday yearlycycledayN
       hourlymean hourlymeanN] = decompose_timeseries(o3(i).sd,  o3
       (i).data);
    % Using hourlmean matrix and smoothing
    hourlymean = naninterp(hourlymean); % first fix NaNs
    hourlymean_anomaly = (hourlymean - m) ./ m * 100; % Compute
       anomaly
    temp = repmat(hourlymean_anomaly,3,3);
    %h=ones(7)/49; % Create filter
    h = ones(3,7)./(3*7);
    smoothtemp= filter2(h, temp); % smooth the extended matrix
    o3anomaly(:,:,i) = smoothtemp(366:366+364,25:25+23); % extract
       original matrix from the smoothed results
    o3anomalyN(:,:,i) = hourlymeanN;
    disp(['Generating␣anomaly␣matrix␣for␣O3,␣station␣number␣'
       num2str(i)]);
end


for i=1:length(pm10)
    m = nanmean(pm10(i).data);
    [dailycycle dailycycleN yearlycycleday yearlycycledayN
       hourlymean hourlymeanN] = decompose_timeseries(pm10(i).sd,
       pm10(i).data);
    % Using hourlmean matrix and smoothing
    hourlymean = naninterp(hourlymean); % first fix NaNs
    hourlymean_anomaly = (hourlymean - m) ./ m * 100; % Compute
       anomaly
    temp = repmat(hourlymean_anomaly,3,3);
    %h=ones(7)/49; % Create filter
    h = ones(3,7)./(3*7);
    smoothtemp= filter2(h, temp); % smooth the extended matrix
    pm10anomaly(:,:,i) = smoothtemp(366:366+364,25:25+23); % extract
        original matrix from the smoothed results
    pm10anomalyN(:,:,i) = hourlymeanN;
    disp(['Generating␣anomaly␣matrix␣for␣PM10,␣station␣number␣'
       num2str(i)]);
end

for i=1:length(pm25)
    m = nanmean(pm25(i).data);
    [dailycycle dailycycleN yearlycycleday yearlycycledayN
       hourlymean hourlymeanN] = decompose_timeseries(pm25(i).sd,
       pm25(i).data);
    % Using hourlmean matrix and smoothing
    hourlymean = naninterp(hourlymean); % first fix NaNs
    hourlymean_anomaly = (hourlymean - m) ./ m * 100; % Compute
       anomaly
    temp = repmat(hourlymean_anomaly,3,3);
    %h=ones(7)/49; % Create filter
    h = ones(3,7)./(3*7);
    smoothtemp= filter2(h, temp); % smooth the extended matrix
```

```
    pm25anomaly(:,:,i) = smoothtemp(366:366+364,25:25+23); % extract
        original matrix from the smoothed results
    pm25anomalyN(:,:,i) = hourlymeanN;
    disp(['Generating anomaly matrix for PM2.5, station number '
       num2str(i)]);
end
```

Listing 5 – The Matlab script `importHourlyRawAirbase.m` which is used for reading the hourly observations from the raw AirBase dataset.

```
function [sd data] = importHourlyRawAirbase(file)

[raw meta]= readtext(file, '\t', '', '');

values = cell2mat(raw(1:end,2:2:end));
qualflag = cell2mat(raw(1:end,3:2:end));
values(qualflag < 1) = NaN;

data = reshape(transpose(values), numel(values), 1);

%date = nan(size(values,1) * 24,1);
sd = [];
for i=1:size(values,1)
    sd = [sd; [datenum(raw{i,1}) + datenum(0,0,0,0:23,0,0)]'];
end



end
```

Listing 6 – The Matlab script `decompose_timeseries.m` which is used for calculating the long-term averages in daily and seasonal cycles from the raw AirBase data at background stations in Norway.

```
function [dailycycle dailycycleN yearlycycleday yearlycycledayN
   hourlymean hourlymeanN] = decompose_timeseries(sd, data)

[year month day hour minute second] = datevec(sd);
hours = 0:23;
for i=1:length(hours)
    ind = find(hour == hours(i));
    dailycycle(i) = nanmean(data(ind));
    dailycyclestd(i) = nanstd(data(ind));
    dailycycleN(i) = sum(~isnan(data(ind)));
end

months = 1:12;
for i = 1:length(months)
    ind = find(month == months(i));
    yearlycycle(i) = nanmean(data(ind));
    yearlycyclestd(i) = nanstd(data(ind));
    yearlycycleN(i) = sum(~isnan(data(ind))) / 30 / 24; % number of
       months
end

days = 1:365;
doy = datevec2doy(datevec(sd));
for i = 1:length(days)
    ind = find(doy == days(i));
    yearlycycleday(i) = nanmean(data(ind));
    yearlycycledaystd(i) = nanstd(data(ind));
    yearlycycledayN(i) = sum(~isnan(data(ind))) / 24;      % number
       of days
```

```
end

for i=1:length(days)
    for j=1:length(hours)
        ind = find(doy == days(i) & hour == hours(j));
        if ~isempty(ind)
            hourlymean(i,j) = nanmean(data(ind));
            hourlymeanN(i,j) = sum(~isnan(data(ind)));
        else
            hourlymean(i,j) = NaN;
            hourlymeanN(i,j) = 0;
        end
    end
end

end
```

A.4 Combination of spatial and temporal component

Listing 7 – Main script for combining the spatial and temporal component of the system: BackgroundAtlas_ComponentCombination.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%% COMBINATION OF COMPONENTS %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

gridlat = lat_no;
gridlon = lon_no;
[nRows nCols] = size(gridlat);

for row=1:nRows
    for col=1:nCols
        %d = distance([no2(1).station_latitude_deg]', [no2(1).
            station_longitude_deg(1)]', gridlat, gridlon);
        d = distance(gridlat(row,col), gridlon(row,col), [no2.
            station_latitude_deg]', [no2.station_longitude_deg]');
        [minval minind] = min(d);
        station_no2(row,col) = minind;
        d = distance(gridlat(row,col), gridlon(row,col), [o3.
            station_latitude_deg]', [o3.station_longitude_deg]');
        [minval minind] = min(d);
        station_o3(row,col) = minind;
        d = distance(gridlat(row,col), gridlon(row,col), [pm10.
            station_latitude_deg]', [pm10.station_longitude_deg]');
        [minval minind] = min(d);
        station_pm10(row,col) = minind;
        d = distance(gridlat(row,col), gridlon(row,col), [pm25.
            station_latitude_deg]', [pm25.station_longitude_deg]');
        [minval minind] = min(d);
        station_pm25(row,col) = minind;
    end
    disp(row)
end


%% Generate the final datatsets with the data for all hours and all
   locations

% NO2
doy = reshape(repmat([1:365],24,1), 365*24, 1);
time = repmat([0:23]', 365, 1);
```

```
a=1;
for row=1:nRows
    for col=1:nCols
        % Go through each pixel
        if ~isnan(no2_avg(row,col))
            linind = sub2ind(size(no2anomaly), doy, time+1, ones(
                size(doy))*station_no2(row,col));
            meta_no2(a,:) = [gridlat(row,col) gridlon(row,col)
                no2_avg(row,col)];
            final_no2(a,:) = [no2_avg(row,col) + no2_avg(row,col) *
                no2anomaly(linind) ./ 100]';
            a=a+1;
        end
    end
    disp(row)
end



% O3
a=1;
for row=1:nRows
    for col=1:nCols
        % Go through each pixel
        if ~isnan(o3_avg(row,col))
            linind = sub2ind(size(o3anomaly), doy, time+1, ones(size
                (doy))*station_o3(row,col));
            meta_o3(a,:) = [gridlat(row,col) gridlon(row,col) o3_avg
                (row,col)];
            final_o3(a,:) = [o3_avg(row,col) + o3_avg(row,col) *
                o3anomaly(linind) ./ 100]';
            a=a+1;
        end
    end
    disp(row)
end


% PM10
a=1;
for row=1:nRows
    for col=1:nCols
        % Go through each pixel
        if ~isnan(pm10_avg(row,col))
            linind = sub2ind(size(pm10anomaly), doy, time+1, ones(
                size(doy))*station_pm10(row,col));
            meta_pm10(a,:) = [gridlat(row,col) gridlon(row,col)
                pm10_avg(row,col)];
            final_pm10(a,:) = [pm10_avg(row,col) + pm10_avg(row,col)
                 * pm10anomaly(linind) ./ 100]';
            a=a+1;
        end
    end
    disp(row)
end


% PM25
a=1;
for row=1:nRows
    for col=1:nCols
        % Go through each pixel
        if ~isnan(pm25_avg(row,col))
```

```
                linind = sub2ind(size(pm25anomaly), doy, time+1, ones(
                    size(doy))*station_pm25(row,col));
                meta_pm25(a,:) = [gridlat(row,col) gridlon(row,col)
                    pm25_avg(row,col)];
                final_pm25(a,:) = [pm25_avg(row,col) + pm25_avg(row,col)
                     * pm25anomaly(linind) ./ 100]';
                a=a+1;
            end
        end
        disp(row)
end
```

A.5 Data export

Listing 8 – Main script for exporting the background information: `BackgroundAtlas_Export.m`

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%% DATA EXPORT %%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

outputdirectory = [rootfolder 'Output\'];
cd(outputdirectory);

%load([rootfolder 'InputData\AuxiliaryData\uncertainties.mat'])


% Uncertainties
unc_no2 = 72.9462;
unc_o3 = 33.7579;
unc_pm10 = 96.1341;
unc_pm25 = 88.1881;
final_se_no2 = ones(size(final_no2,1), 1) * unc_no2;
final_se_o3 = ones(size(final_o3, 1), 1) * unc_o3;
final_se_pm10 = ones(size(final_pm10, 1), 1) * unc_pm10;
final_se_pm25 = ones(size(final_pm25, 1), 1) * unc_pm25;

%% Write out background datasets as Netcdf
[nRows nCols] = size(final_no2);
file = 'no2.nc';
nccreate(file,'Latitude','Dimensions',{'rows' nRows},'Format','
   classic');
nccreate(file,'Longitude','Dimensions',{'rows' nRows},'Format','
   classic');
nccreate(file, 'Annual_mean_NO2', 'Dimensions', {'rows', nRows}, '
   Format','classic');
nccreate(file, 'Day_of_year', 'Dimensions', {'columns', nCols}, '
   Format','classic');
nccreate(file, 'Hour_of_day', 'Dimensions', {'columns', nCols}, '
   Format','classic');
nccreate(file,'NO2','Dimensions',{'rows' nRows 'columns' nCols}, '
   Format','classic');
nccreate(file,'Uncertainty', 'Dimensions', {'rows', nRows}, 'Format'
   , 'classic');
ncwrite(file,'Latitude', meta_no2(:,1));
ncwrite(file,'Longitude', meta_no2(:,2));
ncwrite(file,'Annual_mean_NO2', meta_no2(:,3));
ncwrite(file,'Day_of_year', doy');
ncwrite(file,'Hour_of_day', time');
ncwrite(file,'NO2', final_no2);
```

```
ncwrite(file,'Uncertainty', final_se_no2);
ncwriteatt(file, '/', 'Description', ['Estimated␣mean␣background␣
    concentration␣of␣NO2␣for␣' num2str(nRows) '␣locations␣over␣
    Norway␣given␣for␣all␣8760␣hours␣in␣a␣typical␣year']);
ncwriteatt(file, '/', 'Creation_Date', datestr(now));


[nRows nCols] = size(final_o3);
file = 'o3.nc';
nccreate(file,'Latitude','Dimensions',{'rows' nRows},'Format','
    classic');
nccreate(file,'Longitude','Dimensions',{'rows' nRows},'Format','
    classic');
nccreate(file, 'Annual_mean_O3', 'Dimensions', {'rows', nRows}, '
    Format','classic');
nccreate(file, 'Day_of_year', 'Dimensions', {'columns', nCols}, '
    Format','classic');
nccreate(file, 'Hour_of_day', 'Dimensions', {'columns', nCols}, '
    Format','classic');
nccreate(file,'O3','Dimensions',{'rows' nRows 'columns' nCols}, '
    Format','classic');
nccreate(file,'Uncertainty', 'Dimensions', {'rows', nRows}, 'Format'
    , 'classic');
ncwrite(file,'Latitude', meta_o3(:,1));
ncwrite(file,'Longitude', meta_o3(:,2));
ncwrite(file,'Annual_mean_O3', meta_o3(:,3));
ncwrite(file,'Day_of_year', doy');
ncwrite(file,'Hour_of_day', time');
ncwrite(file,'O3', final_o3);
ncwrite(file,'Uncertainty', final_se_o3);
ncwriteatt(file, '/', 'Description', ['Estimated␣mean␣background␣
    concentration␣of␣O3␣for␣' num2str(nRows) '␣locations␣over␣Norway
    ␣given␣for␣all␣8760␣hours␣in␣a␣typical␣year']);
ncwriteatt(file, '/', 'Creation_Date', datestr(now));

[nRows nCols] = size(final_pm10);
file = 'pm10.nc';
nccreate(file,'Latitude','Dimensions',{'rows' nRows},'Format','
    classic');
nccreate(file,'Longitude','Dimensions',{'rows' nRows},'Format','
    classic');
nccreate(file, 'Annual_mean_PM10', 'Dimensions', {'rows', nRows}, '
    Format','classic');
nccreate(file, 'Day_of_year', 'Dimensions', {'columns', nCols}, '
    Format','classic');
nccreate(file, 'Hour_of_day', 'Dimensions', {'columns', nCols}, '
    Format','classic');
nccreate(file,'PM10','Dimensions',{'rows' nRows 'columns' nCols}, '
    Format','classic');
nccreate(file,'Uncertainty', 'Dimensions', {'rows', nRows}, 'Format'
    , 'classic');
ncwrite(file,'Latitude', meta_pm10(:,1));
ncwrite(file,'Longitude', meta_pm10(:,2));
ncwrite(file,'Annual_mean_PM10', meta_pm10(:,3));
ncwrite(file,'Day_of_year', doy');
ncwrite(file,'Hour_of_day', time');
ncwrite(file,'PM10', final_pm10);
ncwrite(file,'Uncertainty', final_se_pm10);
ncwriteatt(file, '/', 'Description', ['Estimated␣mean␣background␣
    concentration␣of␣PM10␣for␣' num2str(nRows) '␣locations␣over␣
    Norway␣given␣for␣all␣8760␣hours␣in␣a␣typical␣year']);
ncwriteatt(file, '/', 'Creation_Date', datestr(now));
```

```
[nRows nCols] = size(final_pm25);
file = 'pm25.nc';
nccreate(file,'Latitude','Dimensions',{'rows' nRows},'Format','
    classic');
nccreate(file,'Longitude','Dimensions',{'rows' nRows},'Format','
    classic');
nccreate(file, 'Annual_mean_PM2.5', 'Dimensions', {'rows', nRows}, '
    Format','classic');
nccreate(file, 'Day_of_year', 'Dimensions', {'columns', nCols}, '
    Format','classic');
nccreate(file, 'Hour_of_day', 'Dimensions', {'columns', nCols}, '
    Format','classic');
nccreate(file,'PM2.5','Dimensions',{'rows' nRows 'columns' nCols}, '
    Format','classic');
nccreate(file,'Uncertainty', 'Dimensions', {'rows', nRows}, 'Format'
    , 'classic');
ncwrite(file,'Latitude', meta_pm25(:,1));
ncwrite(file,'Longitude', meta_pm25(:,2));
ncwrite(file,'Annual_mean_PM2.5', meta_pm25(:,3));
ncwrite(file,'Day_of_year', doy');
ncwrite(file,'Hour_of_day', time');
ncwrite(file,'PM2.5', final_pm25);
ncwrite(file,'Uncertainty', final_se_pm25);
ncwriteatt(file, '/', 'Description', ['Estimated mean background 
    concentration of PM2.5 for ' num2str(nRows) ' locations over 
    Norway given for all 8760 hours in a typical year']);
ncwriteatt(file, '/', 'Creation_Date', datestr(now));




%% Write out the annual mean maps as geotiffs

% NO2
file = 'no2.nc'
lat = ncread(file,'Latitude');
lon = ncread(file,'Longitude');
no2 = ncread(file, 'Annual_mean_NO2');

[longrid latgrid] = meshgrid(0:0.1:35, 55:0.1:75);
[no2grid y x] = gridding2(lat,lon,no2,latgrid,longrid);
no2grid(isnan(no2grid)) = 0;
R = makerefmat(0,55,0.1, 0.1)
%geotiffwrite('test2.tif', test, R)
%[testlat testlon] =meshgrat(griddeddata,R);
no2grid(no2grid==0) = -9999;
geotiffwrite('no2.tif', no2grid, R)
%xlswrite('background.xlsx', flipud(latgrid), 'lat')
%xlswrite('background.xlsx', flipud(longrid), 'lon')
%xlswrite('background.xlsx', flipud(no2grid), 'no2')

% O3
file = 'o3.nc'
lat = ncread(file,'Latitude');
lon = ncread(file,'Longitude');
o3 = ncread(file, 'Annual_mean_O3');

[longrid latgrid] = meshgrid(0:0.1:35, 55:0.1:75);
[o3grid y x] = gridding2(lat,lon,o3,latgrid,longrid);
o3grid(isnan(o3grid)) = -9999;
R = makerefmat(0,55,0.1, 0.1)
%geotiffwrite('test2.tif', test, R)
%[testlat testlon] =meshgrat(griddeddata,R);
```

```
o3grid(o3grid==0) = -9999;
geotiffwrite('o3.tif', o3grid, R)
%xlswrite('background.xlsx', flipud(o3grid), 'o3')


% PM10
file = 'pm10.nc'
lat = ncread(file,'Latitude');
lon = ncread(file,'Longitude');
pm10 = ncread(file, 'Annual_mean_PM10');

[longrid latgrid] = meshgrid(0:0.1:35, 55:0.1:75);
[pm10grid y x] = gridding2(lat,lon,pm10,latgrid,longrid);
pm10grid(isnan(pm10grid)) = -9999;
R = makerefmat(0,55,0.1, 0.1)
pm10grid(pm10grid==0) = -9999;
geotiffwrite('pm10.tif', pm10grid, R)
%xlswrite('background.xlsx', flipud(pm10grid), 'pm10')

% PM25
file = 'pm25.nc'
lat = ncread(file,'Latitude');
lon = ncread(file,'Longitude');
pm25 = ncread(file, 'Annual_mean_PM2.5');

[longrid latgrid] = meshgrid(0:0.1:35, 55:0.1:75);
[pm25grid y x] = gridding2(lat,lon,pm25,latgrid,longrid);
pm25grid(isnan(pm25grid)) = -9999;
R = makerefmat(0,55,0.1, 0.1)
pm25grid(pm25grid==0) = -9999;
geotiffwrite('pm25.tif', pm25grid, R)
%xlswrite('background.xlsx', flipud(pm25grid), 'pm25')
```

A.6 Various auxiliary scripts

Listing 9 – readCSVinStruct.m

```
function data = readCSVinStruct(file, separator)
% Currently assumes only one header line

% Read data
[temp r] = readtext(file, separator);

% Read header
headers = temp(1,:);
nCols = length(headers);

% Create struct based on variable names in header line
for i=1:length(temp)-1
    for j=1:length(headers)
        data(i).(headers{j}) = temp{i+1,j};
    end
end


end
```

Listing 10 – plotStationMatrix.m

```
function ax = plotStationMatrix(sd, data, anomaly)
```

```
m = nanmean(data);
[dailycycle dailycycleN yearlycycleday yearlycycledayN hourlymean
    hourlymeanN] = decompose_timeseries(sd,  data);

cfigure(30,15)
ax(1) = subaxis(4,1,1, 'Margin', 0.05);
    pcolor(1:365,1:24, hourlymean');
    %imagesc(1:365,1:24, hourlymean');
    %set(gca, 'YDir', 'normal');
    %stationname = [data.station_city ' - ' data.station_name];
    datetick
    hcb = colorbar
    %cblabel('Conc. [\mug m^{-3}]')
    hcb.Label.String = 'Conc.␣[\mug␣m^{-3}]';
    xlim([0 365])
    caxis([quantile(hourlymean(:), 0.01) quantile(hourlymean(:),
        0.99)])
ax(2) = subaxis(4,1,2)
    pcolor(1:365,1:24, hourlymeanN');
    hcb = colorbar
    %cblabel('N (samples)')
    hcb.Label.String = 'N␣(samples)';
    caxis([0 15])
    datetick
    xlim([0 365])
ax(3) = subaxis(4,1,3)
    m = nanmean(hourlymean(:));
    pcolor(1:365,1:24, (hourlymean' - m) ./ m .* 100);
    hcb = colorbar
    %cblabel('Anomaly [%]')
    hcb.Label.String = 'Anomaly␣[%]';
    caxis([-100 100])
    set(hcb, 'YTick', [-100 -50 0 50 100])
    datetick
    xlim([0 365])
ax(4) = subaxis(4,1,4)
    pcolor(1:365,1:24, anomaly');
    hcb = colorbar
    %cblabel('Smoothed Anomaly [%]')
    hcb.Label.String = 'Smoothed␣Anomaly␣[%]';
    caxis([-100 100])
    set(hcb, 'YTick', [-100 -50 0 50 100])
    datetick
    xlim([0 365])

sublabel(ax, 0, -10, 'FontName', 'helvetica', 'FontWeight', 'bold',
    'BackgroundColor', 'none', 'FontSize', 8)     ;
suplabel('Time␣of␣day␣[hours]', 'y', [.05 .075 .85 .85])
set(gcf,'Color','w')
colormap(jet)

end
```

Listing 11 – read_emep.m

```
function [latemep lonemep val country] = read_emep(file)
%% Read EMEP file

%file = 'C:\Users\ps\Dropbox\Work\NILU\Data\InSitu\EMEP\Gridded50km\
    EMEP_NOx_2003.txt';

[data, result]= readtext(file, ';', '#', '', '');
data = data(2:end,:);
```

```
i = [data{:,5}]';
j = [data{:,6}]';
val = [data{:,8}]';
country = {data{:,1}}';
[latemep lonemep] = compLLfromEMEP(i, j);


end
```

Listing 12 – compLLfromEMEP.m

```
function [lat lon] = compLLfromEMEP(i, j)
% Computes latitude longitude of pixel center given row and col of
    EMEP
% pixel

xpol = 8;
ypol = 110;
d = 50;
phi0 = pi/3;
R = 6370;
M = R/d * (1 + sin(phi0));
r = sqrt( (i - xpol).^2 + (j - ypol).^2 );
lambda0 = -32;

lat = 90 - 360/pi .* atan(r/M);
%if lat == 90
%    lon = 0;
%else
lon = lambda0 + 180/pi .* atan((i - xpol) ./ (ypol - j));
%end

% Fix North Pole
lon(lat==90) = 0;

% Fix special situations caused by atan() .see PDF PBL_CCE_EmepGrid.
    pdf
quadrant1ind = find(i > xpol & j > ypol);
quadrant2ind = find(i < xpol & j > ypol);
lon(quadrant1ind) = lon(quadrant1ind) + 180;
lon(quadrant2ind) = lon(quadrant2ind) - 180;

%    lon(find(lon<-100)) = lon(find(lon<-100)) + 180; % fix the "
    wrapping"

end
```

B Updated anomaly matrices

For completeness, the following set of figures shows all the updated matrices of anomalies from the long-term mean that were computed at all stations in Norway for all four species of interest in this study. These datasets were computed from the up-to-date version 8 of the European air quality database, AirBase. The database provides data up to and including the year 2013. These anomaly matrices are used to provide an approximate estimate of the annual and daily cycle of the background concentrations.

B.1 Anomaly matrices for NO_2

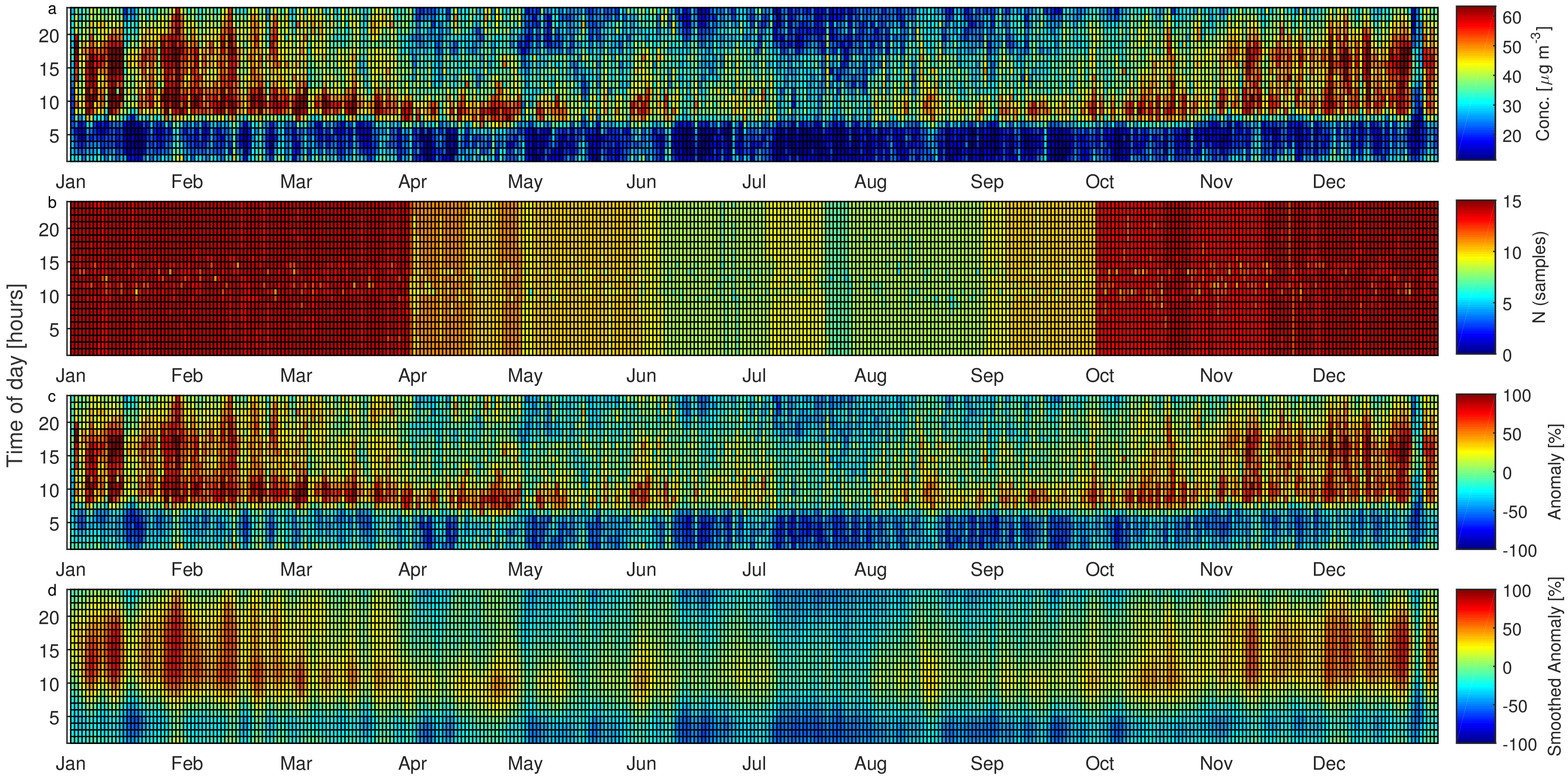

Figure 11 – NO_2 at station *NO0015A Rådhuset*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter. It should be noted that recent visits to this site have cast doubt on the fact if this station truly is representative of urban background. Future work should take this into account.

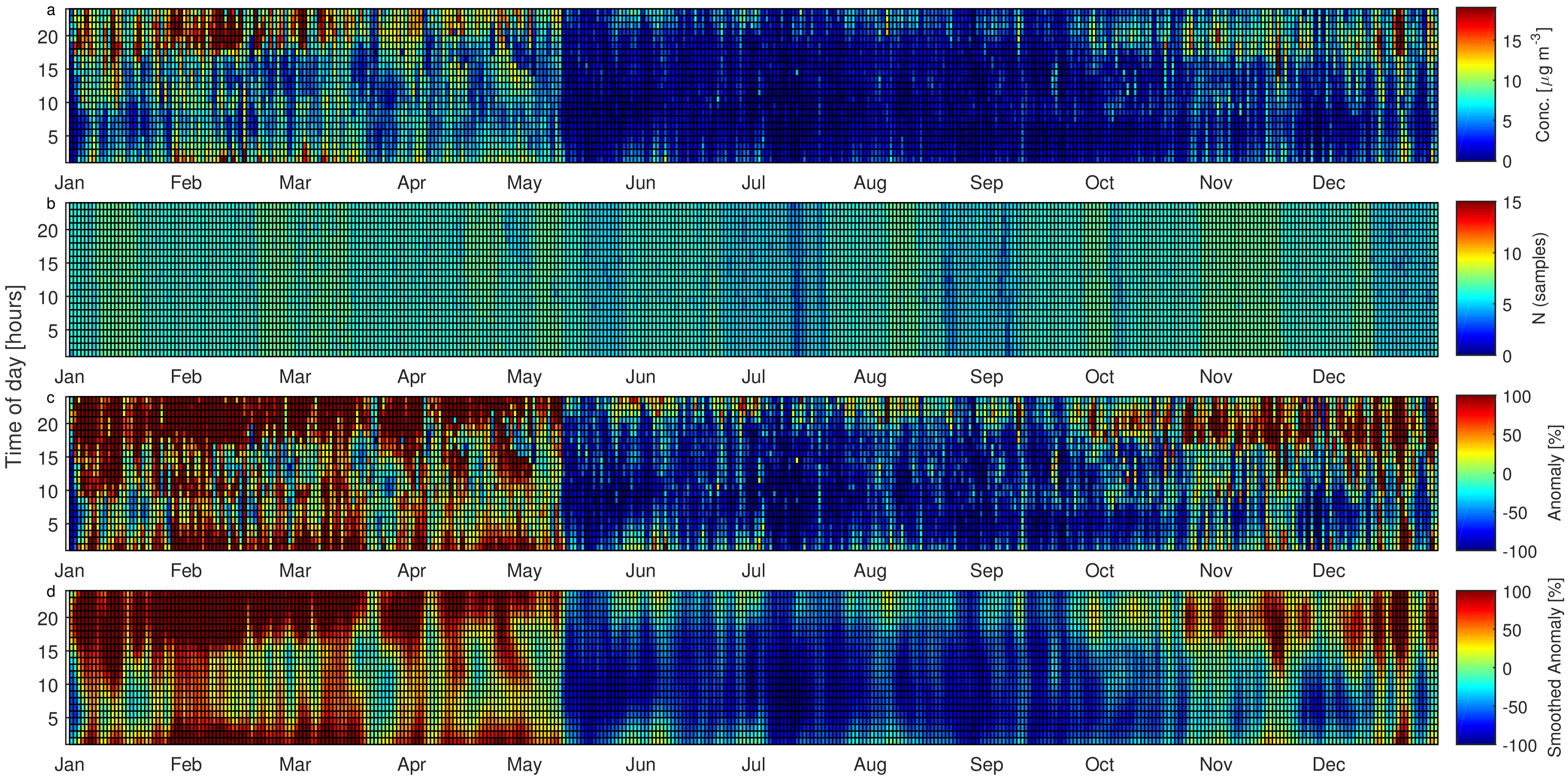

Figure 12 – NO_2 at station *NO0062A Haukenes*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

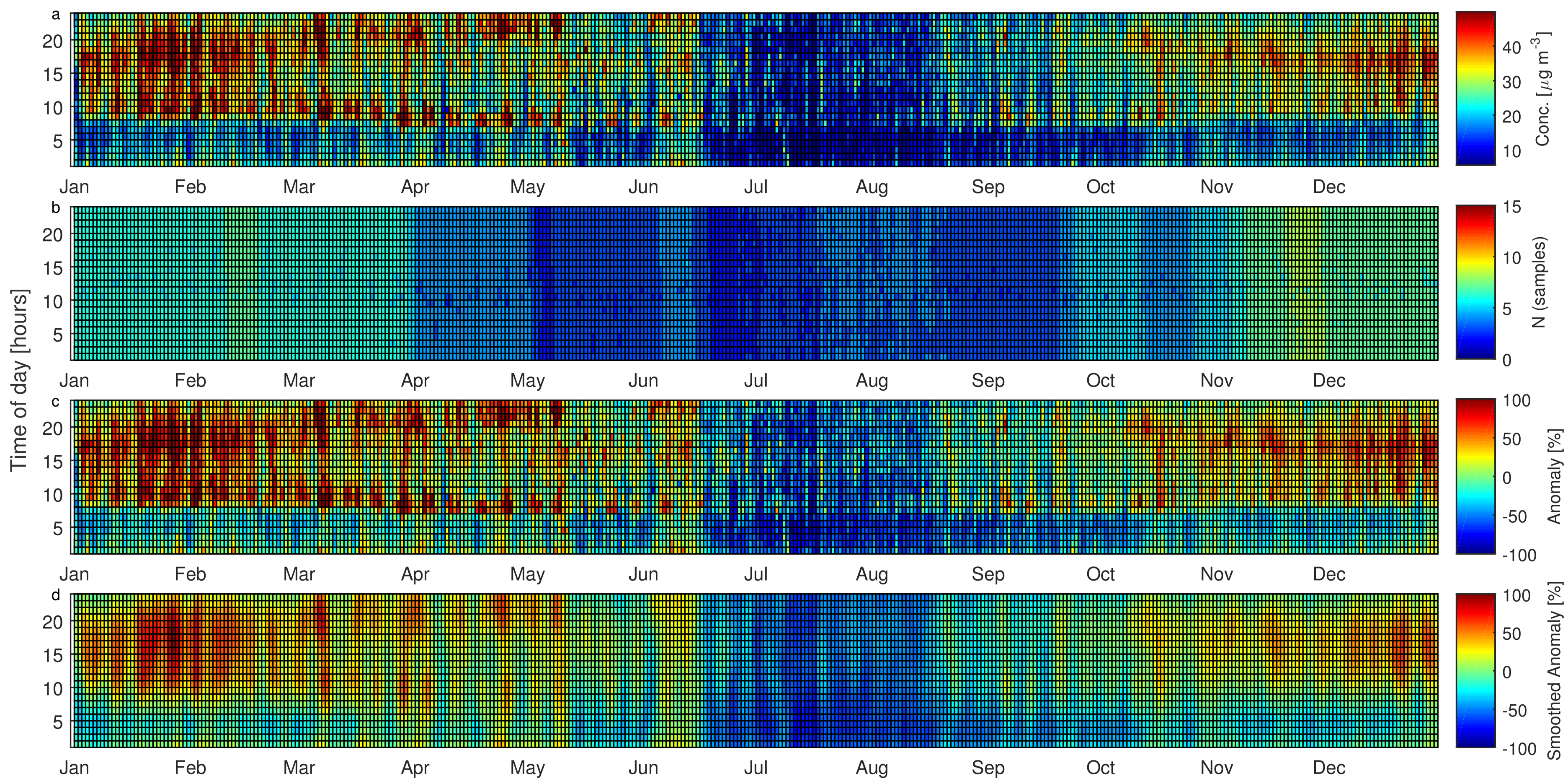

Figure 13 – NO_2 at station *NO0063A Stener Heyerdahl*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

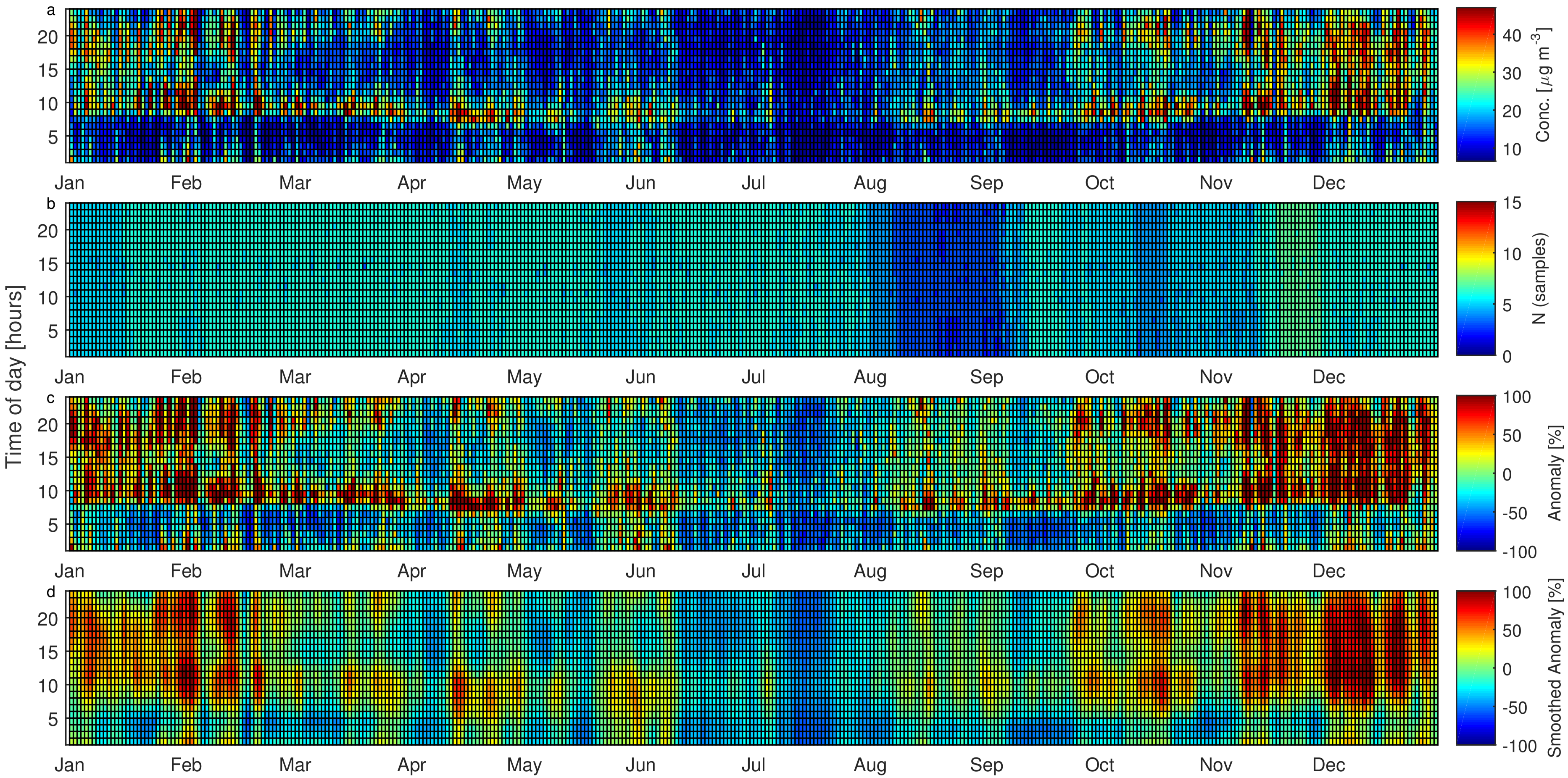

Figure 14 – NO_2 at station *NO0065A Våland*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

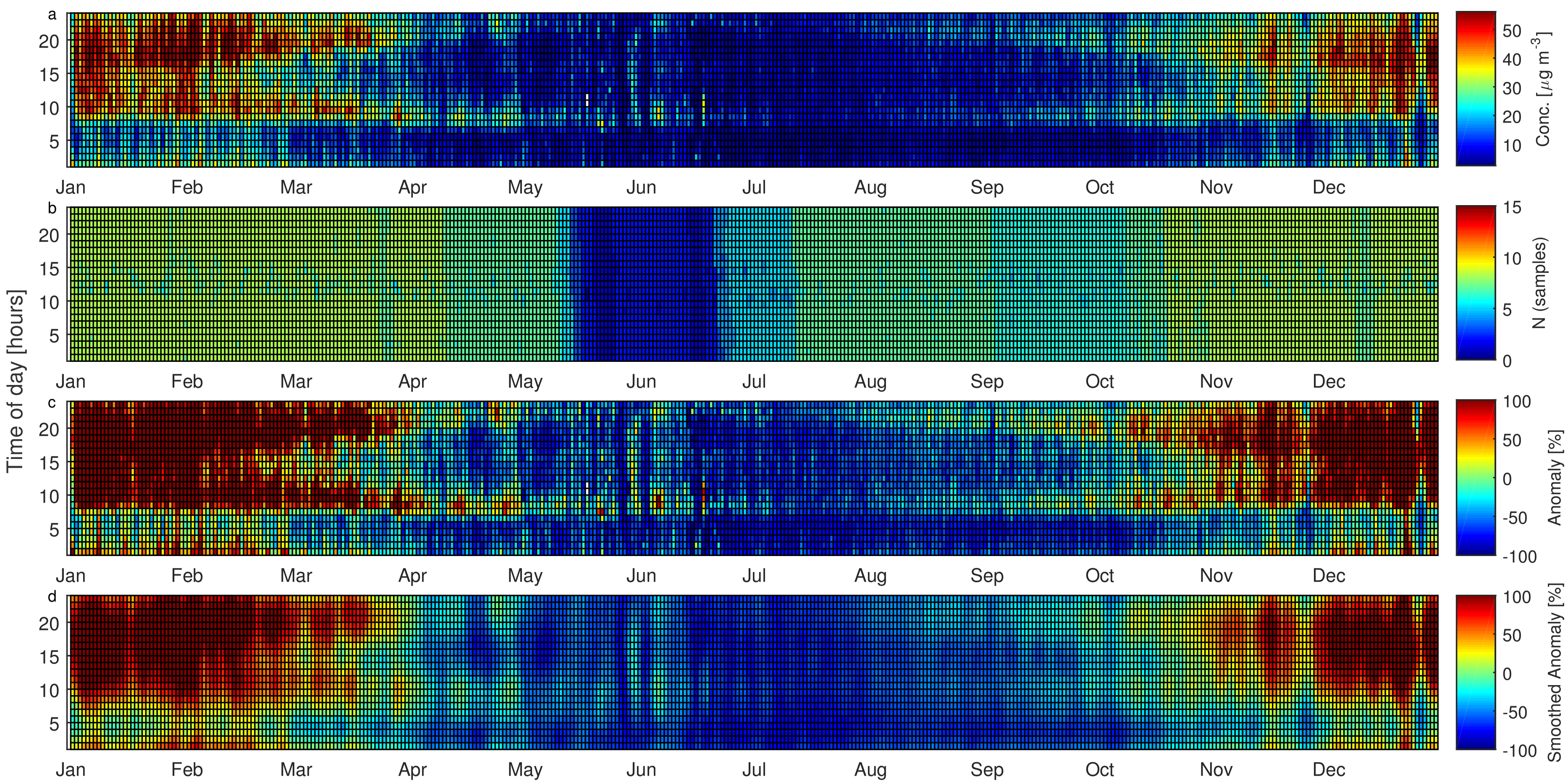

Figure 15 – NO_2 at station *NO0075A Barnehagen*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

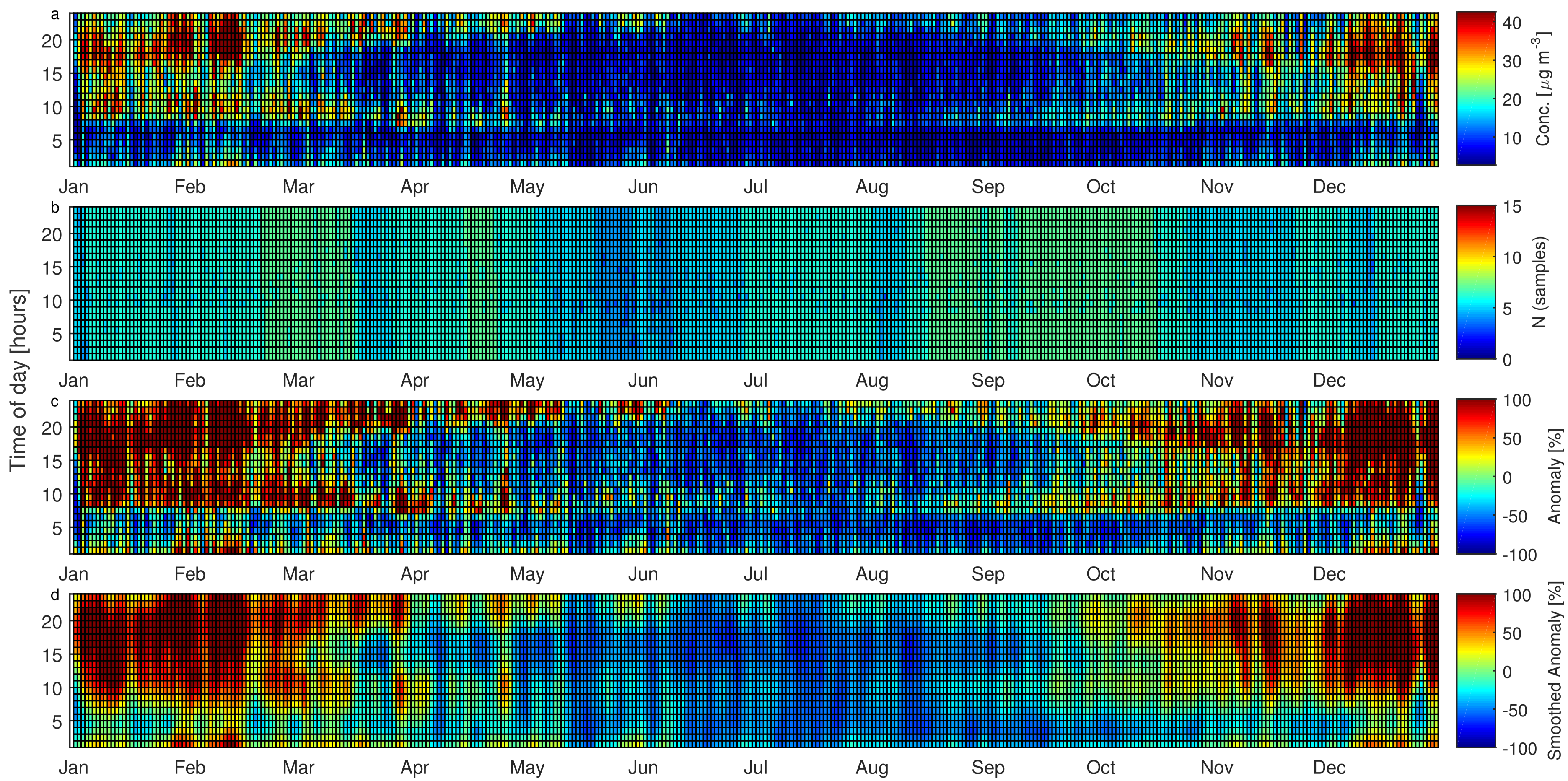

Figure 16 – NO_2 at station *NO0080A Øyekast*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

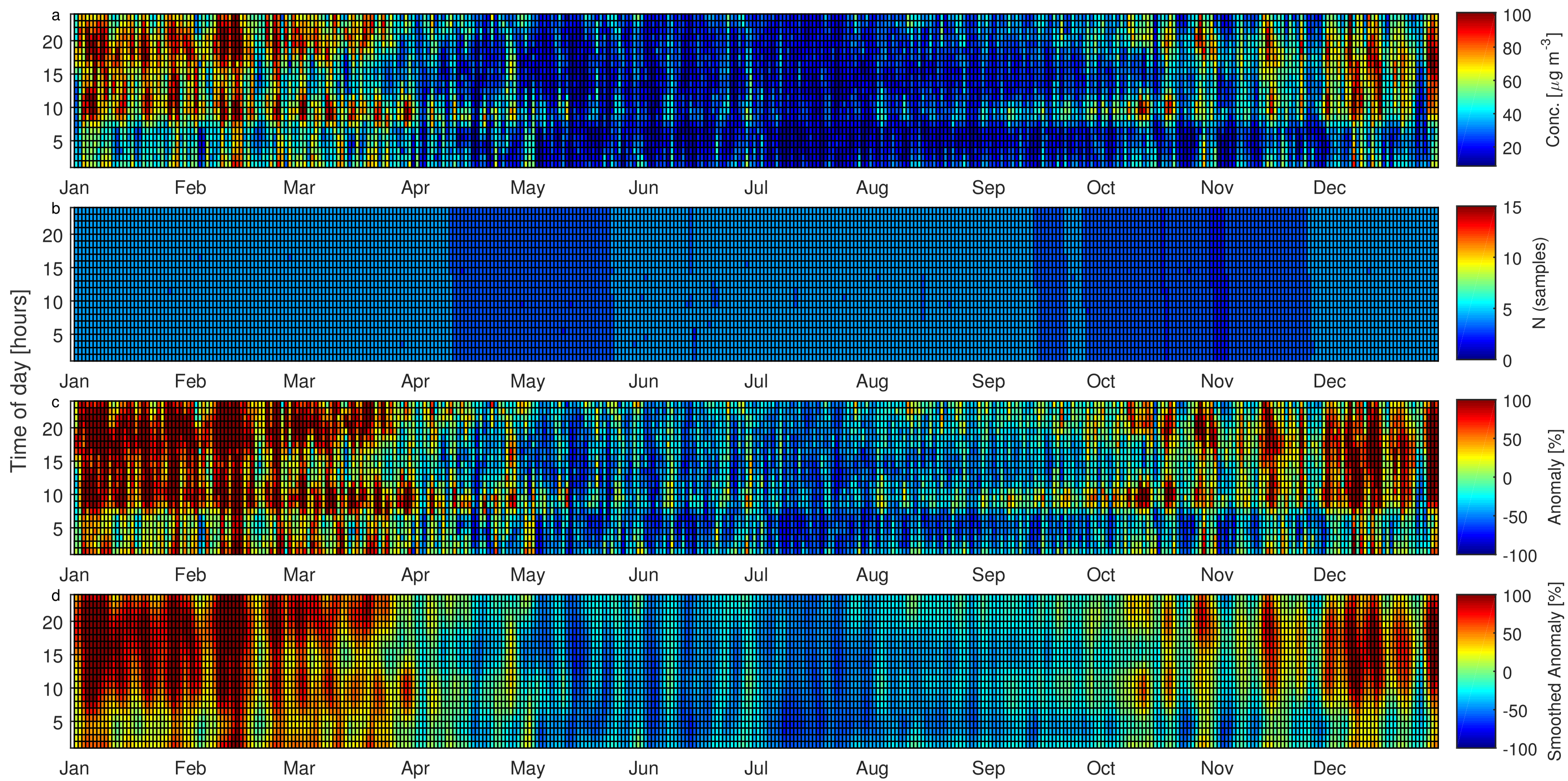

Figure 17 – NO_2 at station *NO0088A Grønland*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

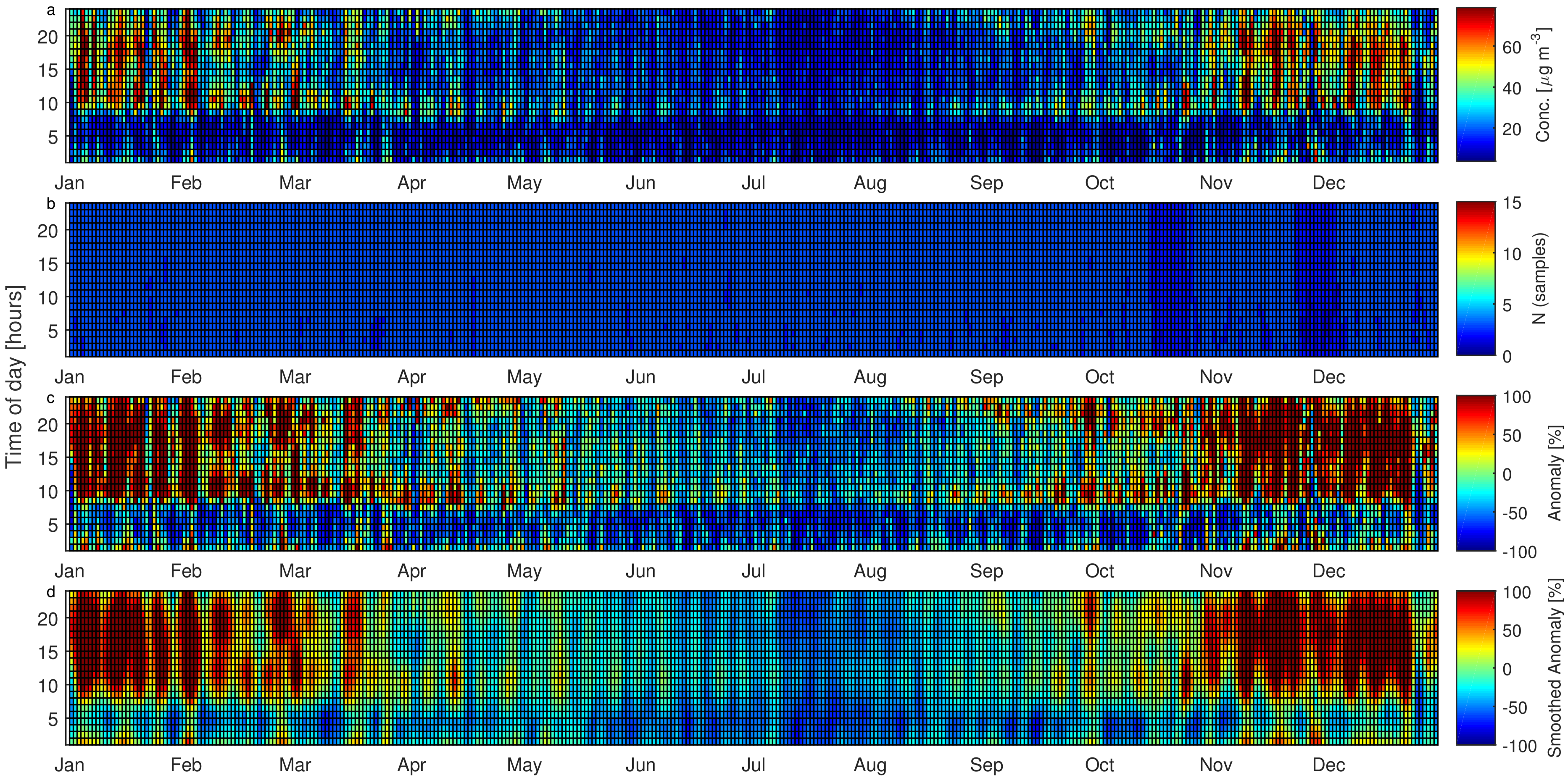

Figure 18 – NO_2 at station *NO0089A Torvet*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

B.2 Anomaly matrices for O_3

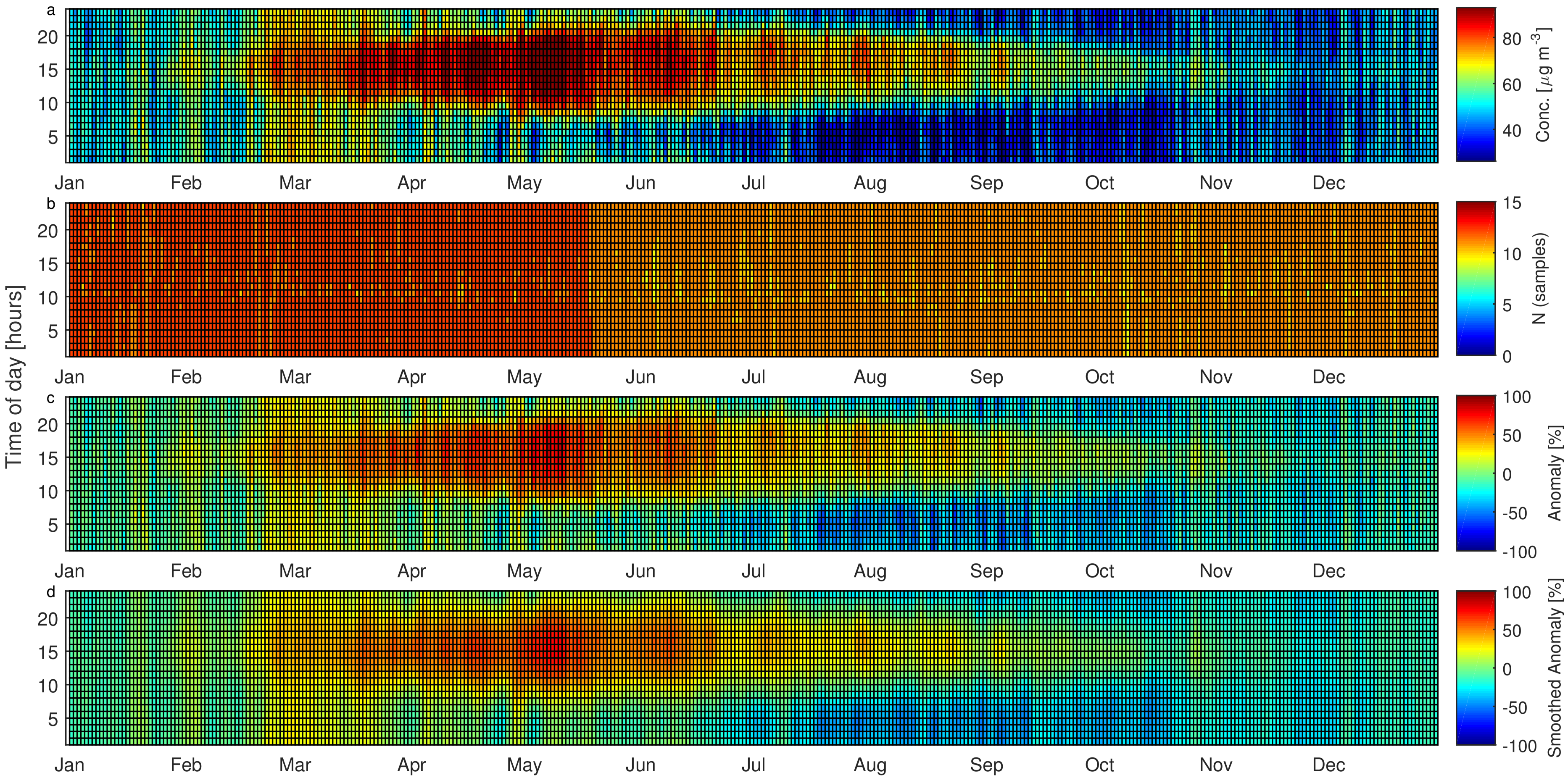

Figure 19 – O_3 at station *NO0001R Birkenes*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

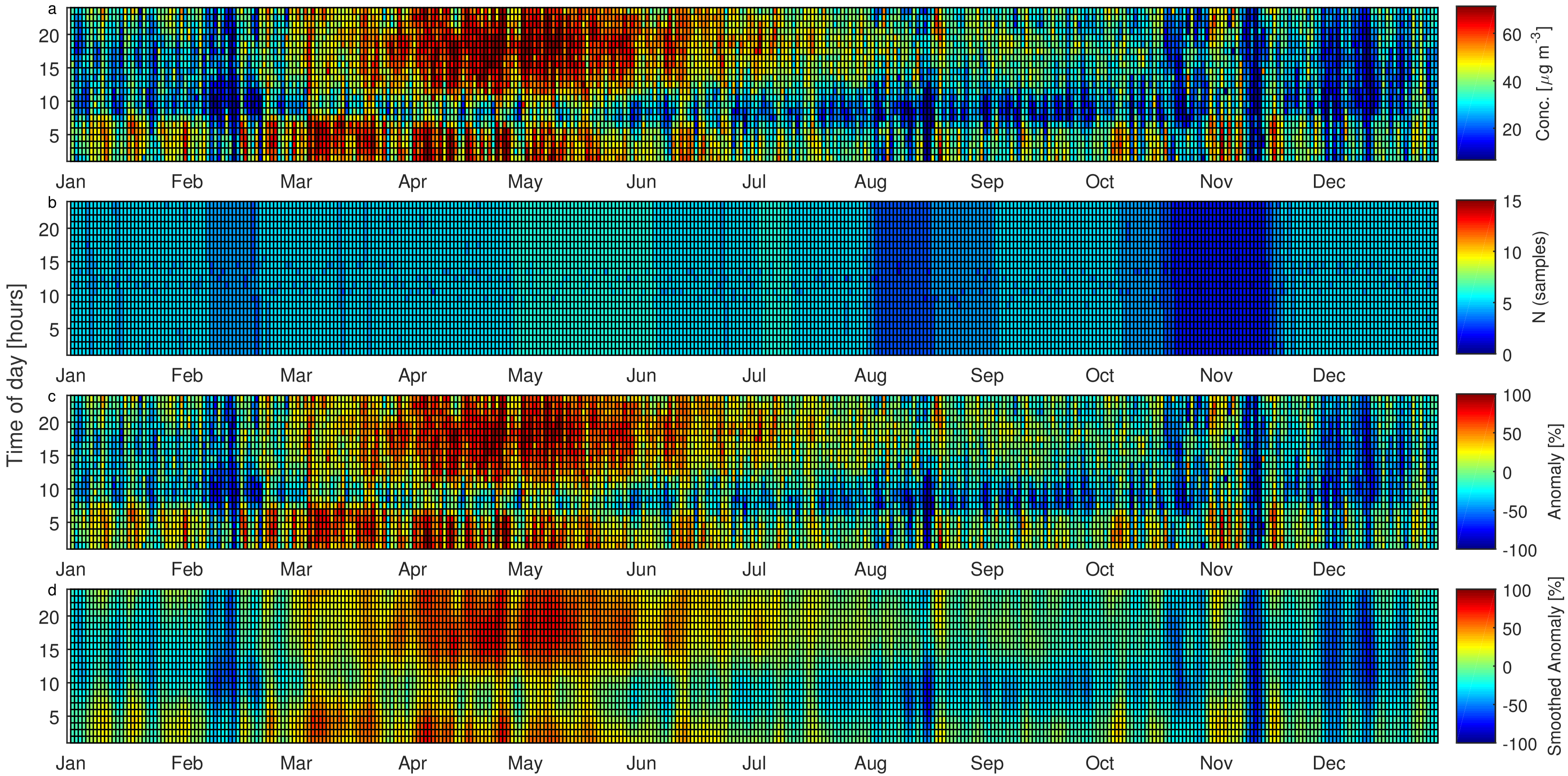

Figure 20 – O_3 at station *NO0015A Rådhuset*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter. It should be noted that recent visits to this site have cast doubt on the fact if this station truly is representative of urban background. Future work should take this into account.

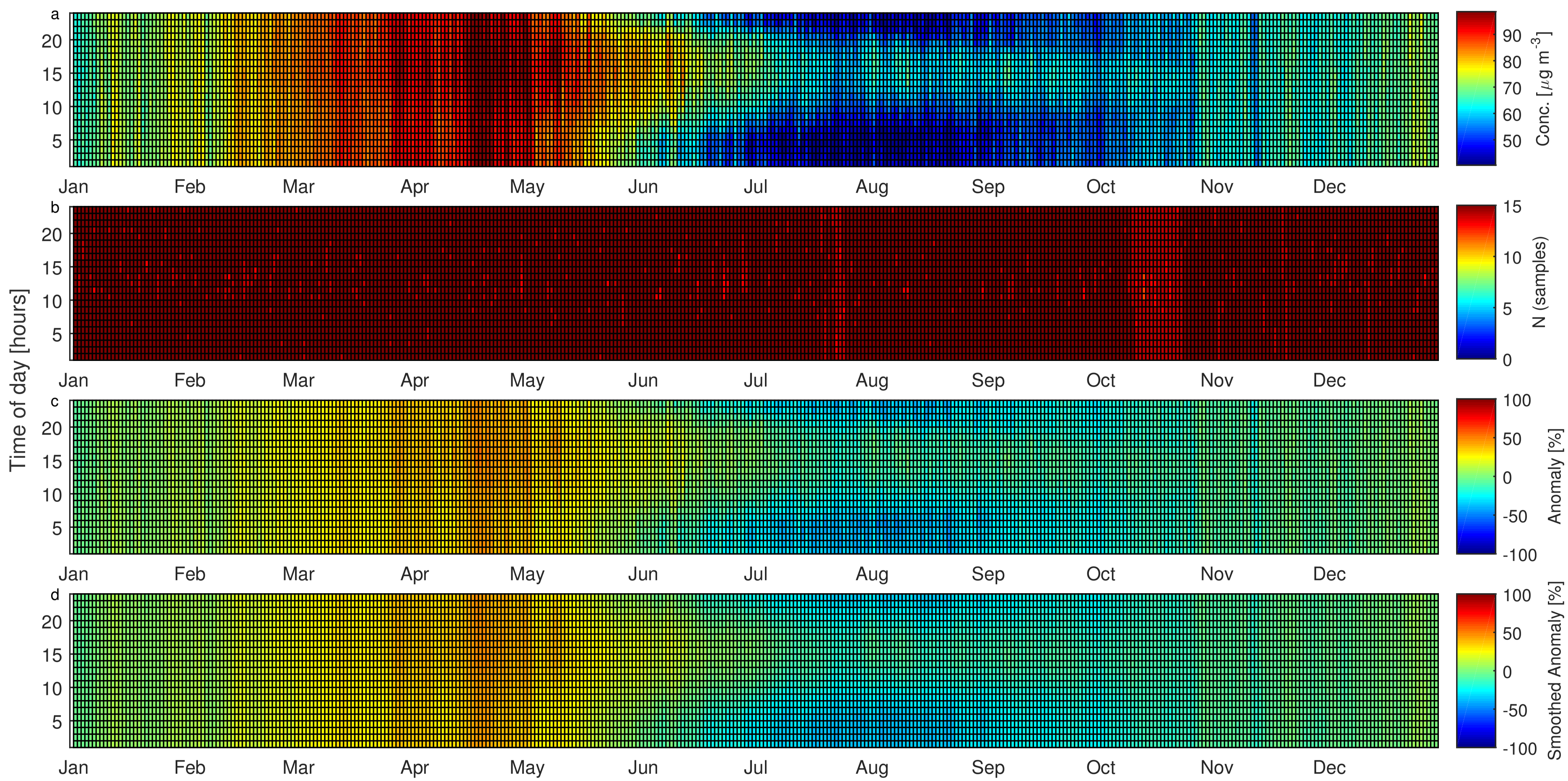

Figure 21 – O_3 at station *NO0015R Tustervatn*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

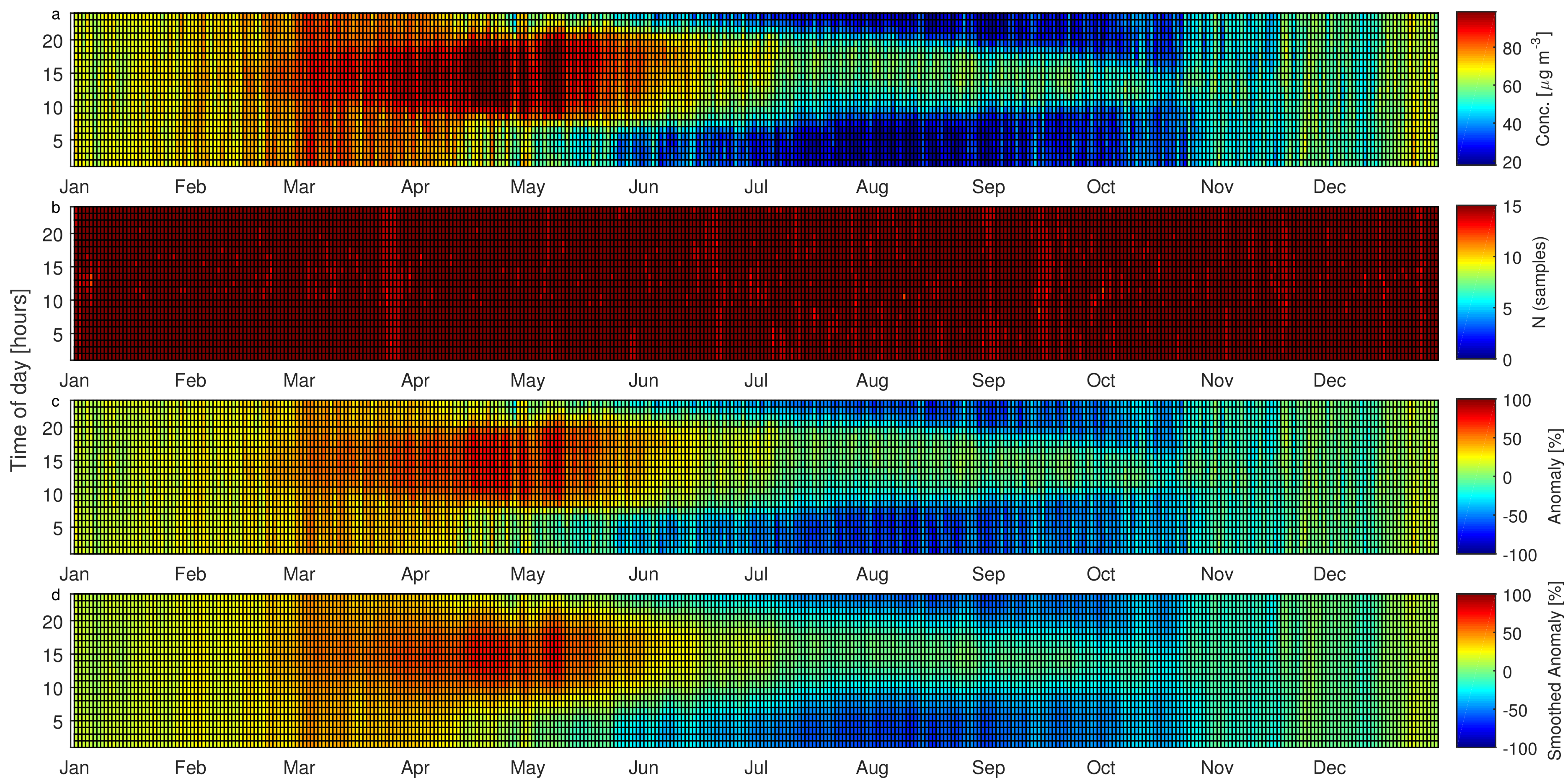

Figure 22 – O_3 at station *NO0039R Kårvatn*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

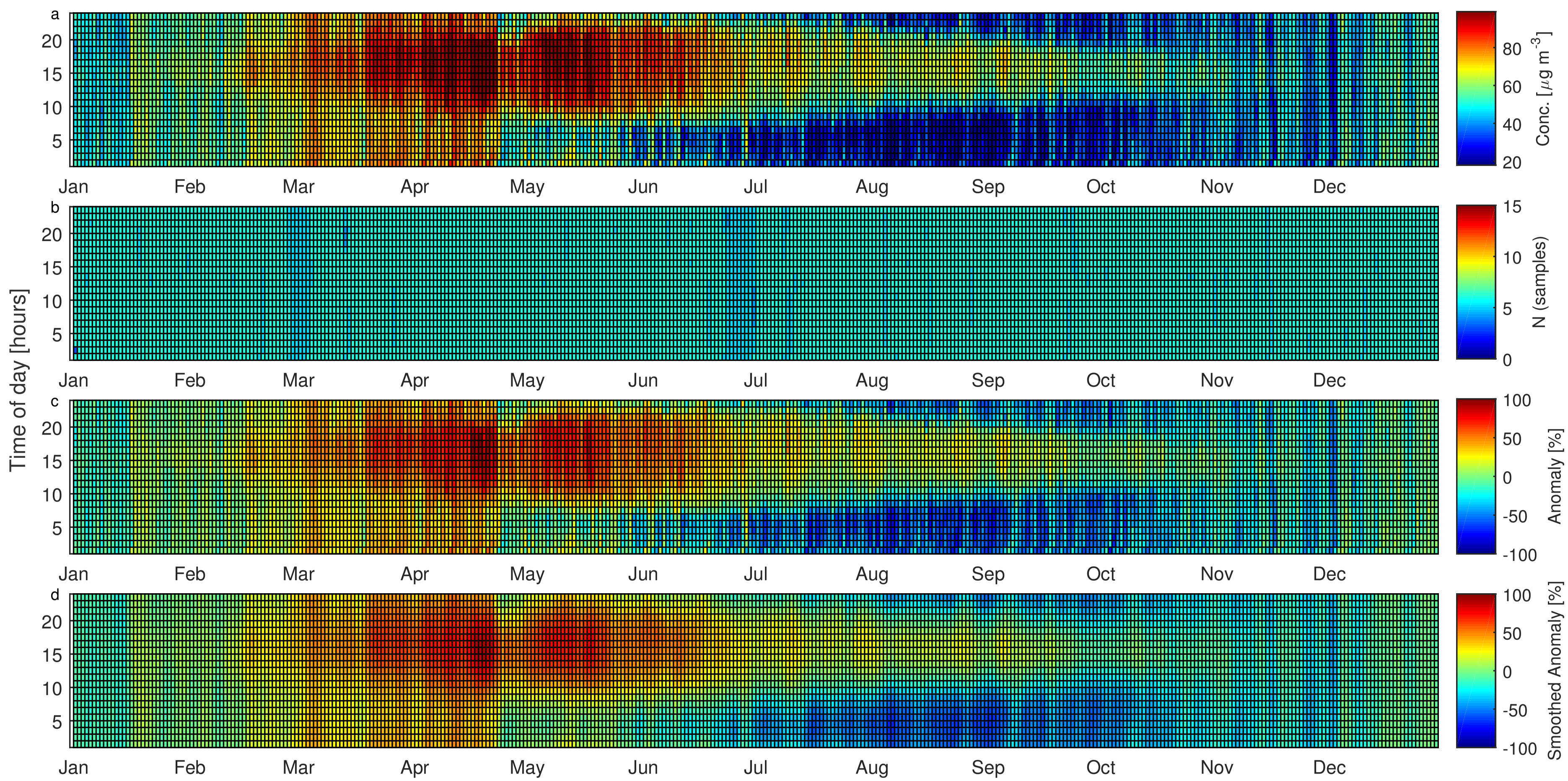

Figure 23 – O_3 at station *NO0041R Osen*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

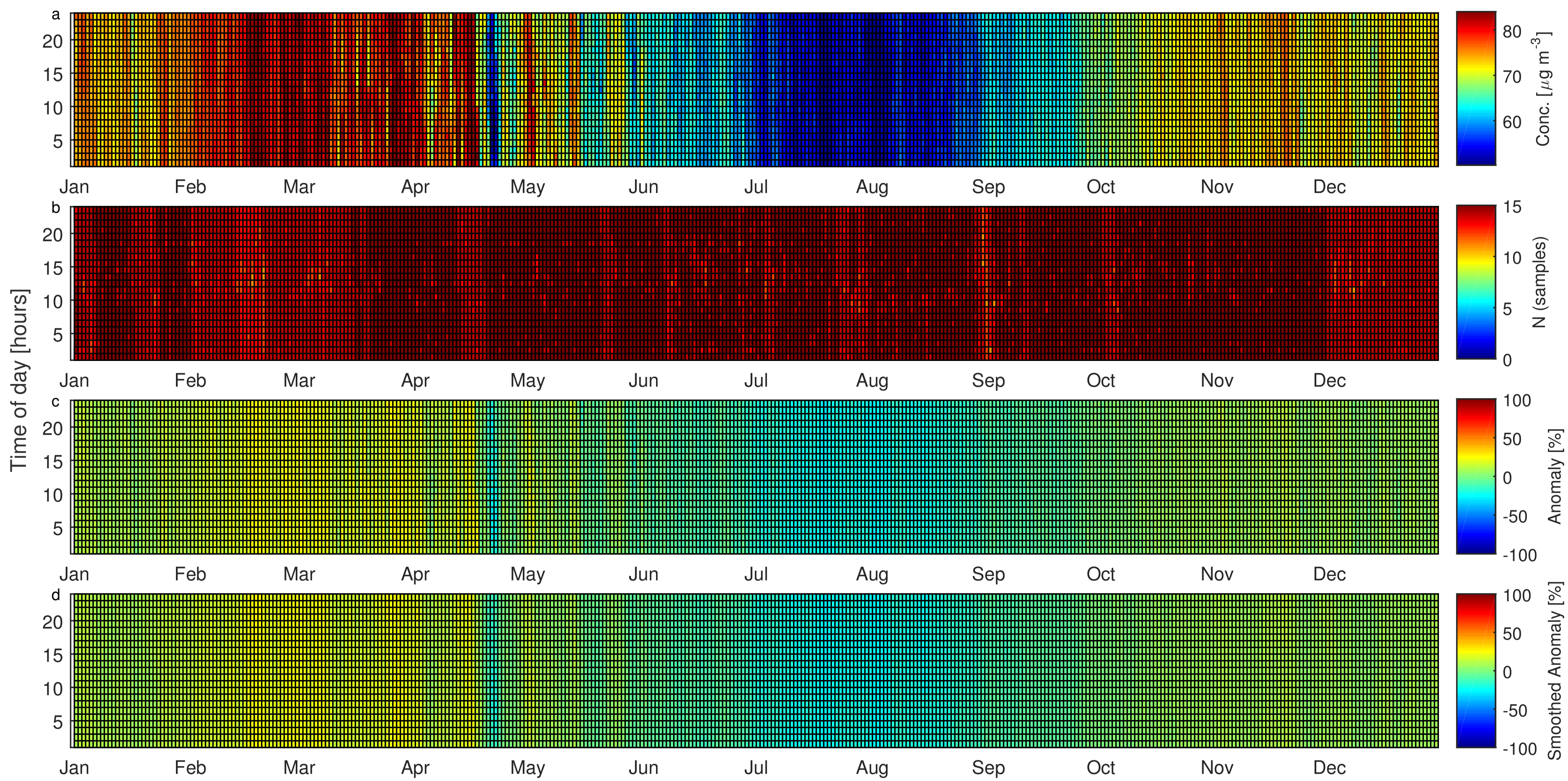

Figure 24 – O_3 at station *NO0042R Zeppelin*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

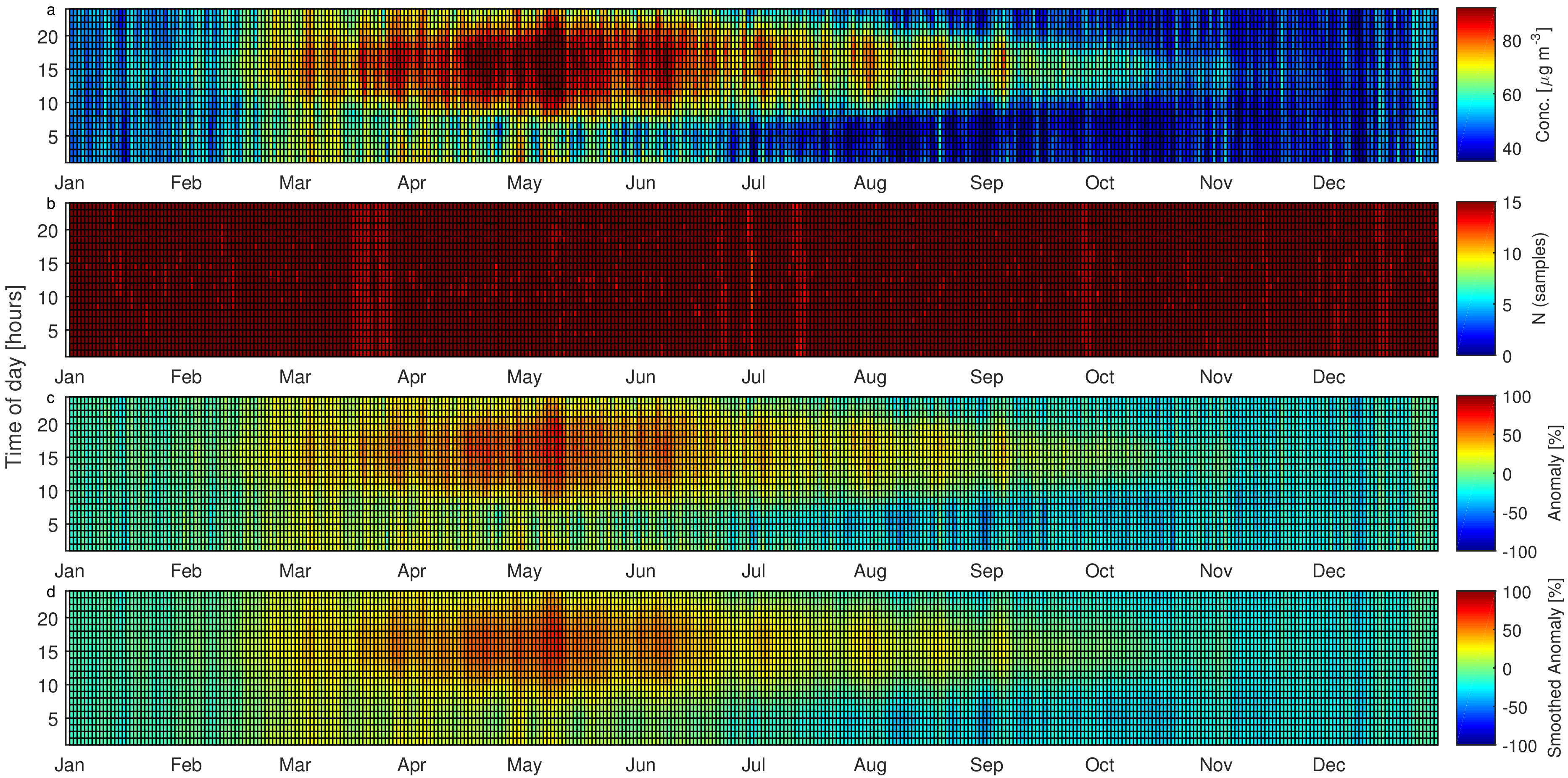

Figure 25 – O_3 at station *NO0043R Prestebakke*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

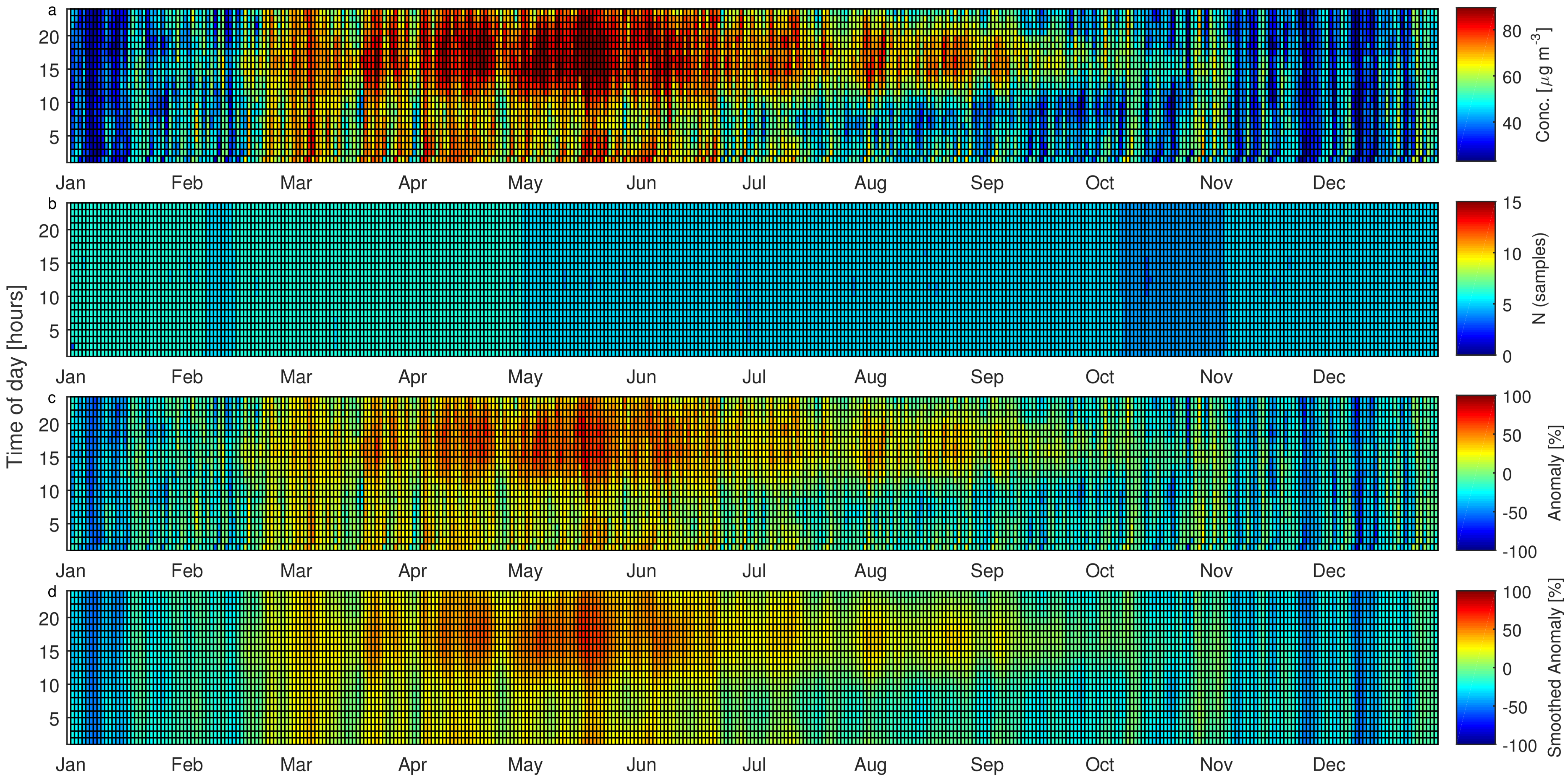

Figure 26 – O_3 at station *NO0045R Jeløya*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

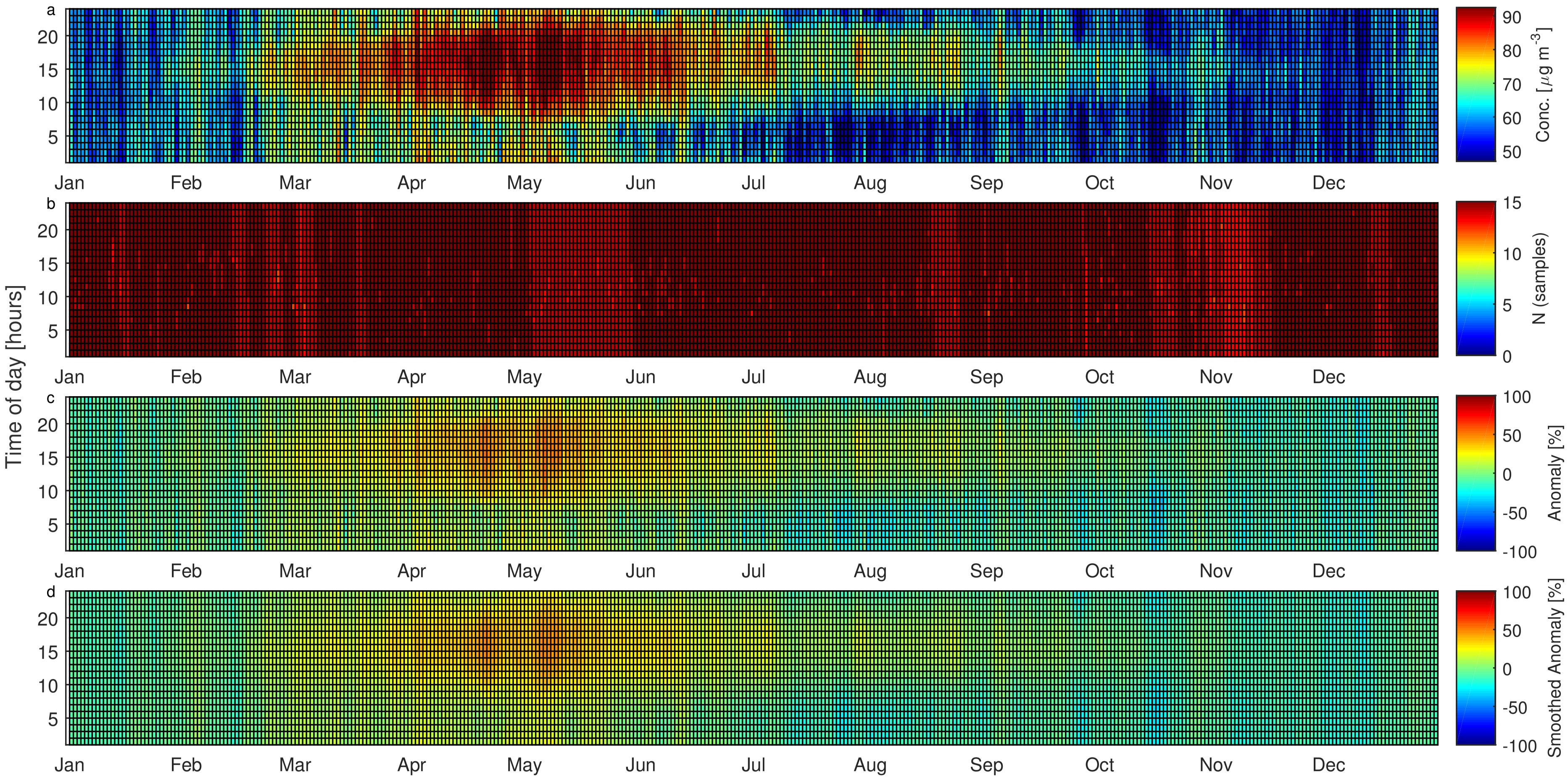

Figure 27 – O_3 at station *NO0052R Sandve*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

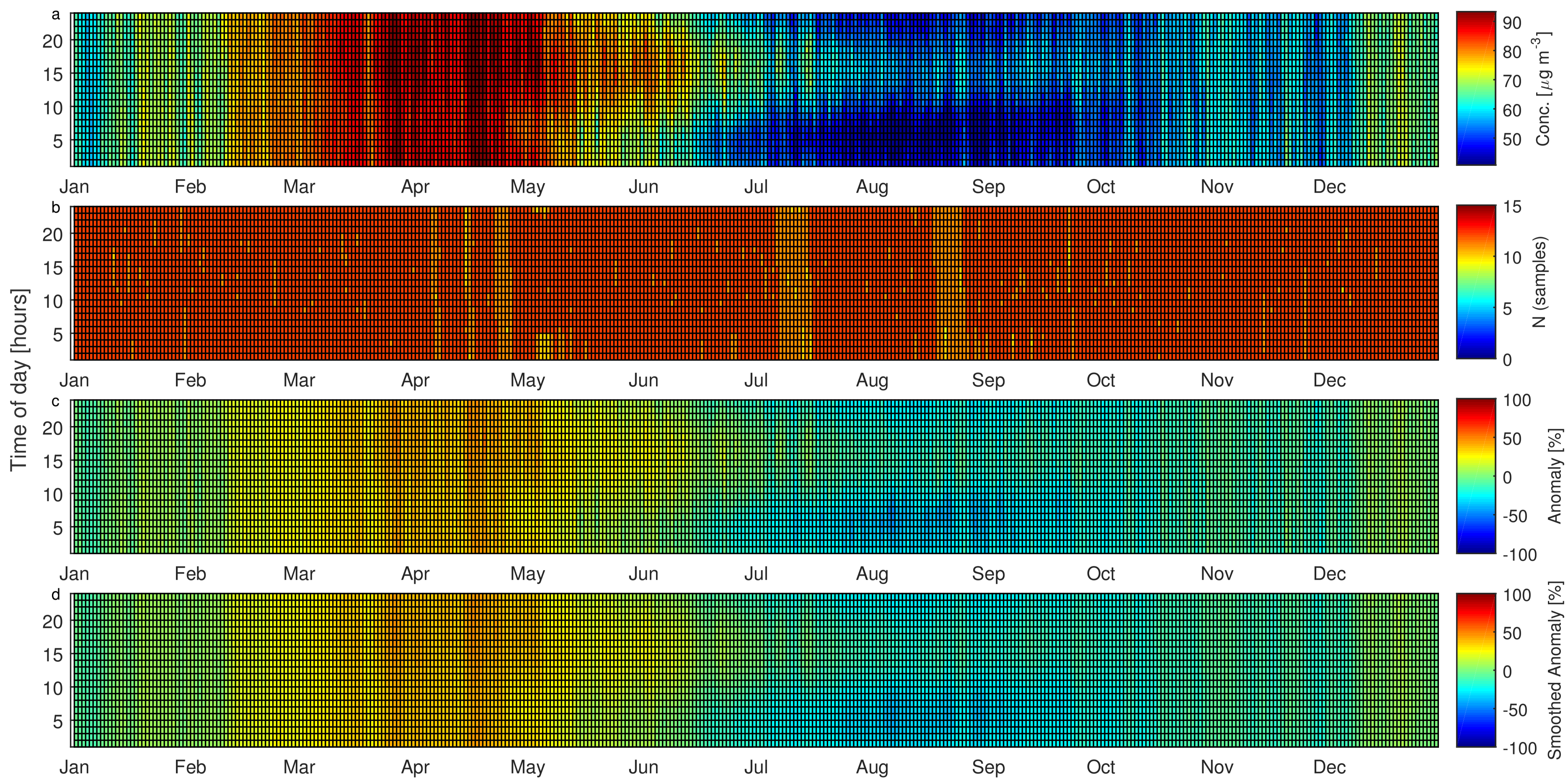

Figure 28 – O_3 at station *NO0055R Karasjok*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

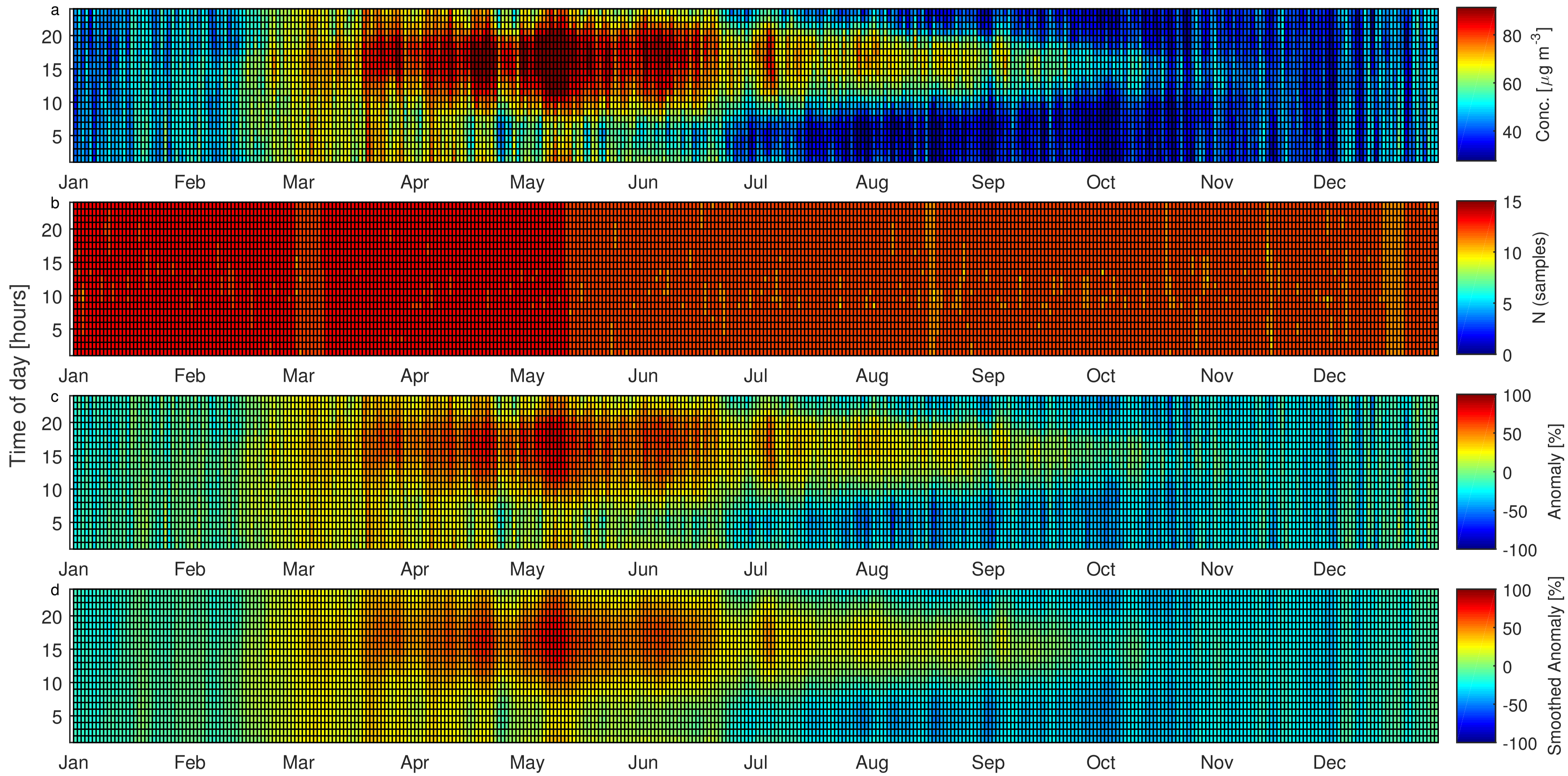

Figure 29 – O_3 at station *NO0056R Hurdal*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

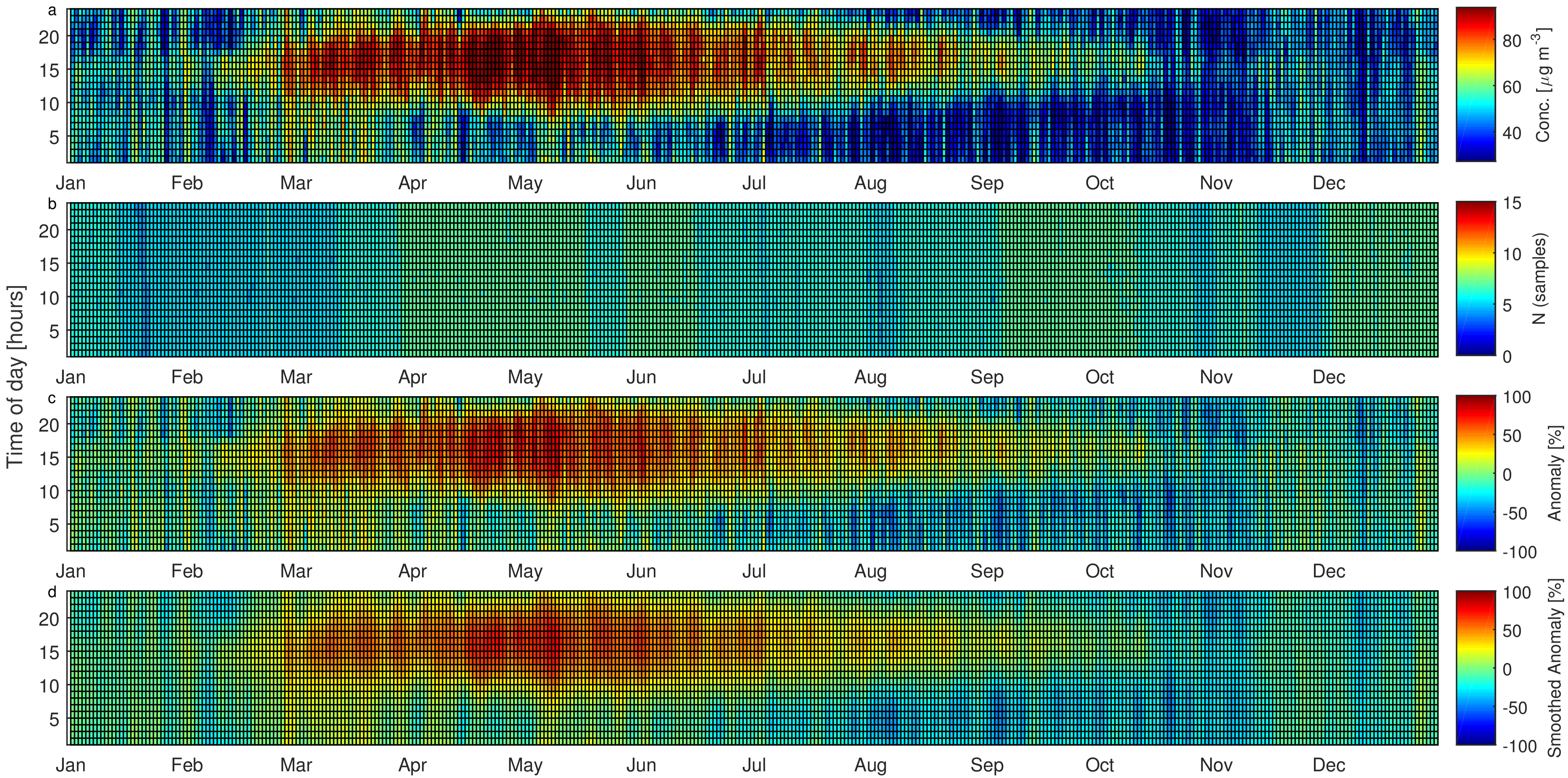

Figure 30 – O_3 at station *NO0062A Haukenes*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

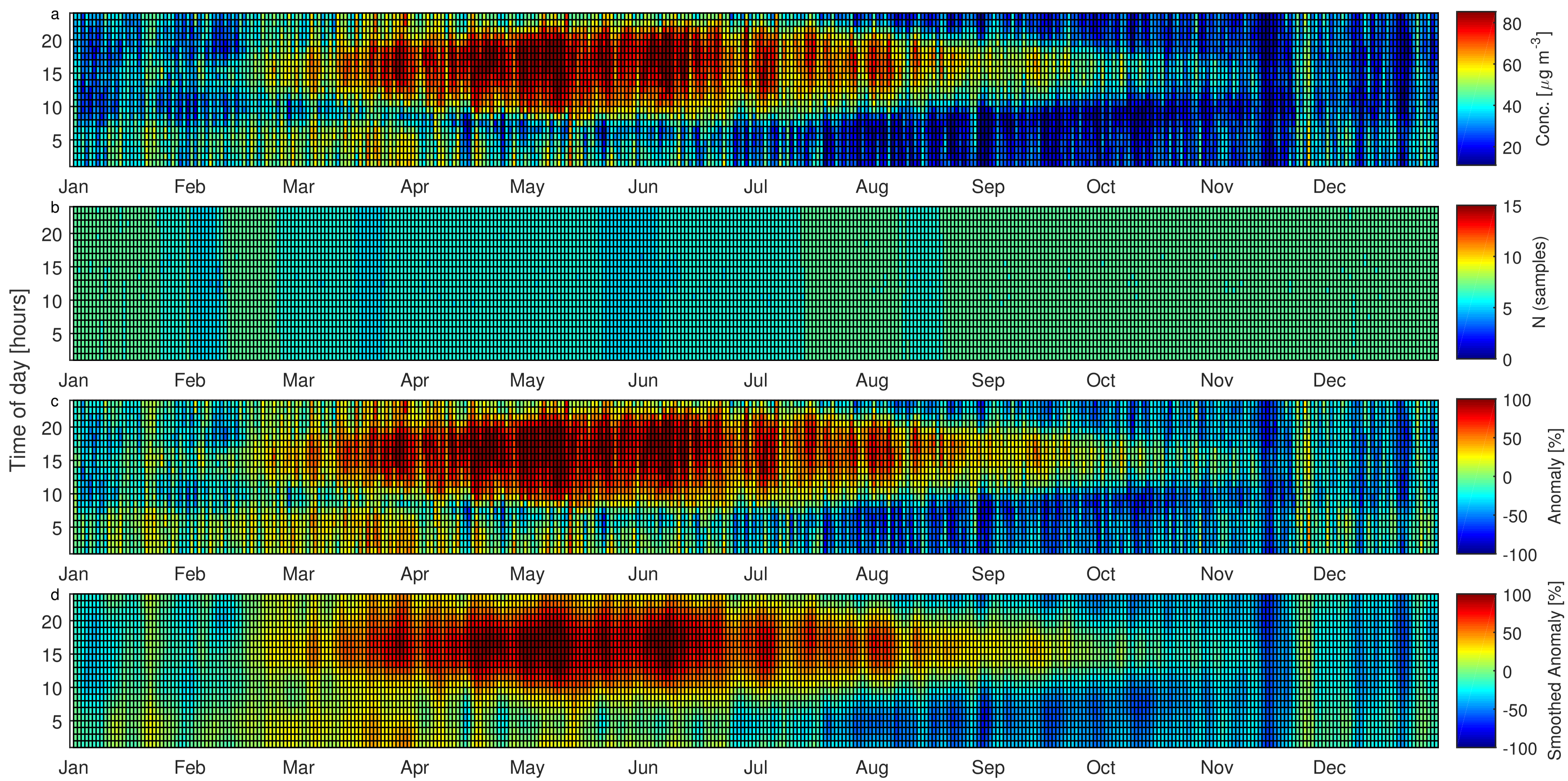

Figure 31 – O_3 at station *NO0081A Bærum*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

B.3 Anomaly matrices for PM_{10}

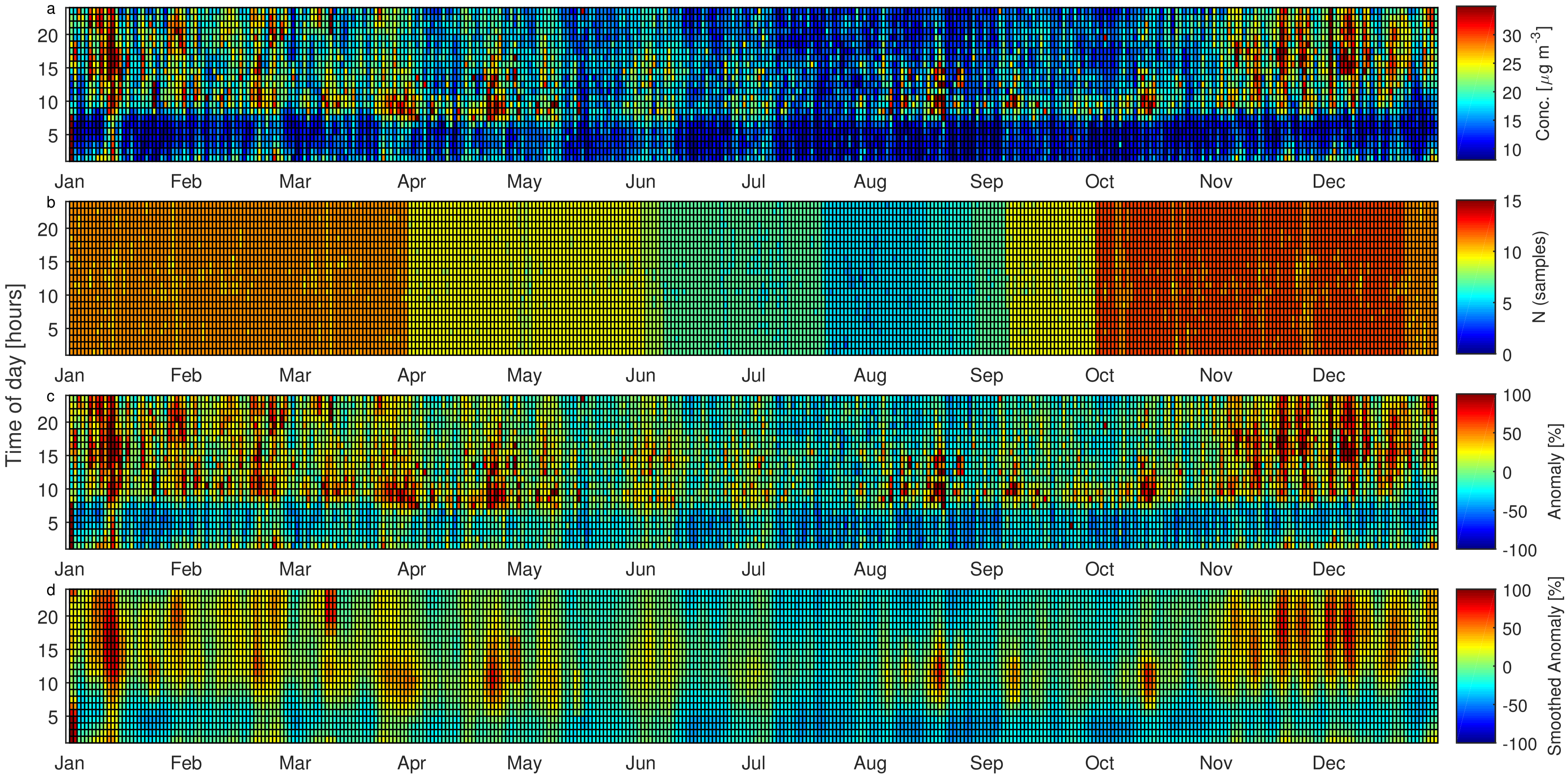

Figure 32 – PM_{10} at station *NO0015A Rådhuset*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter. It should be noted that recent visits to this site have cast doubt on the fact if this station truly is representative of urban background. Future work should take this into account.

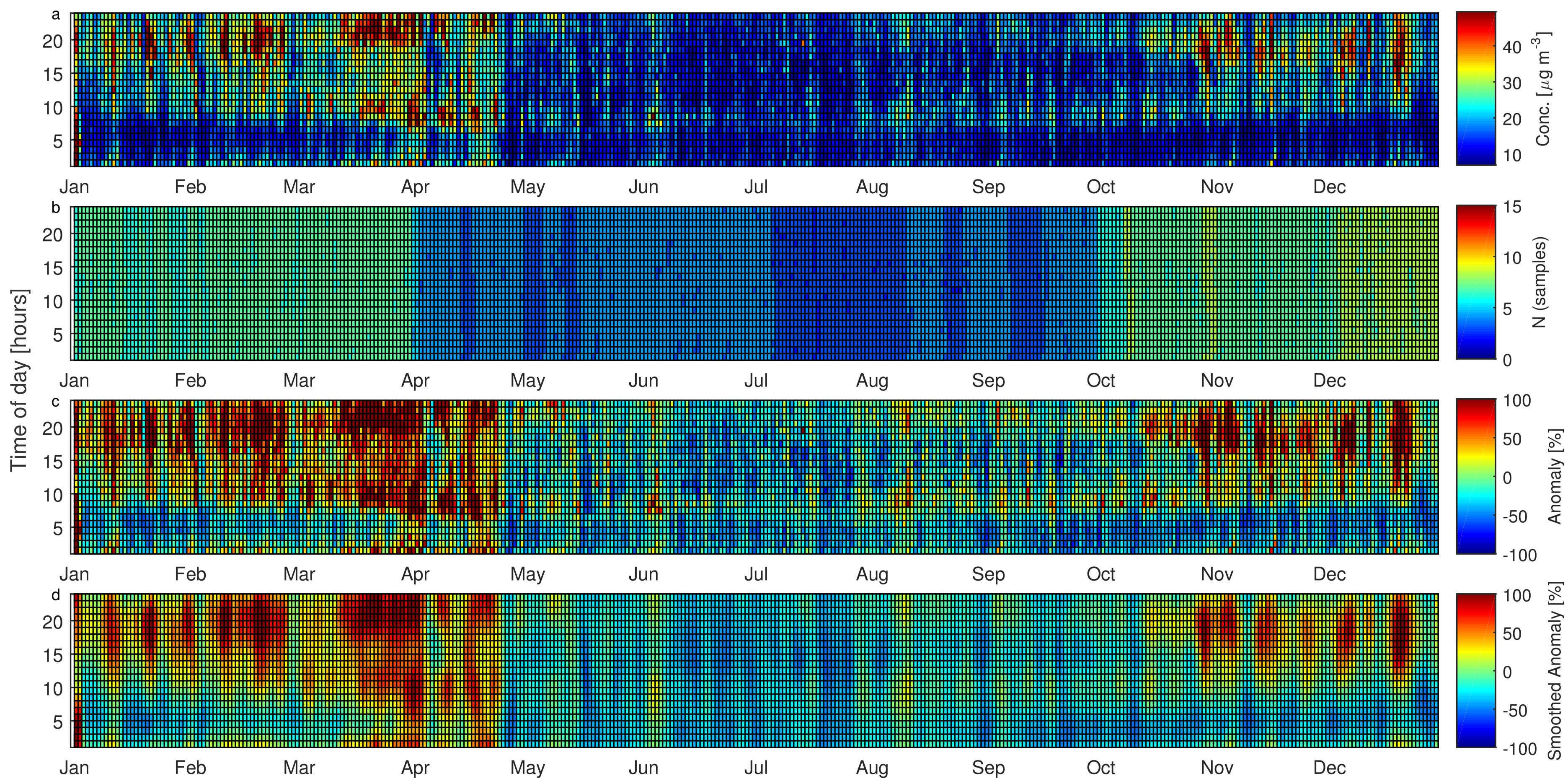

Figure 33 – PM_{10} at station *NO0016A Nedre Storgate*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

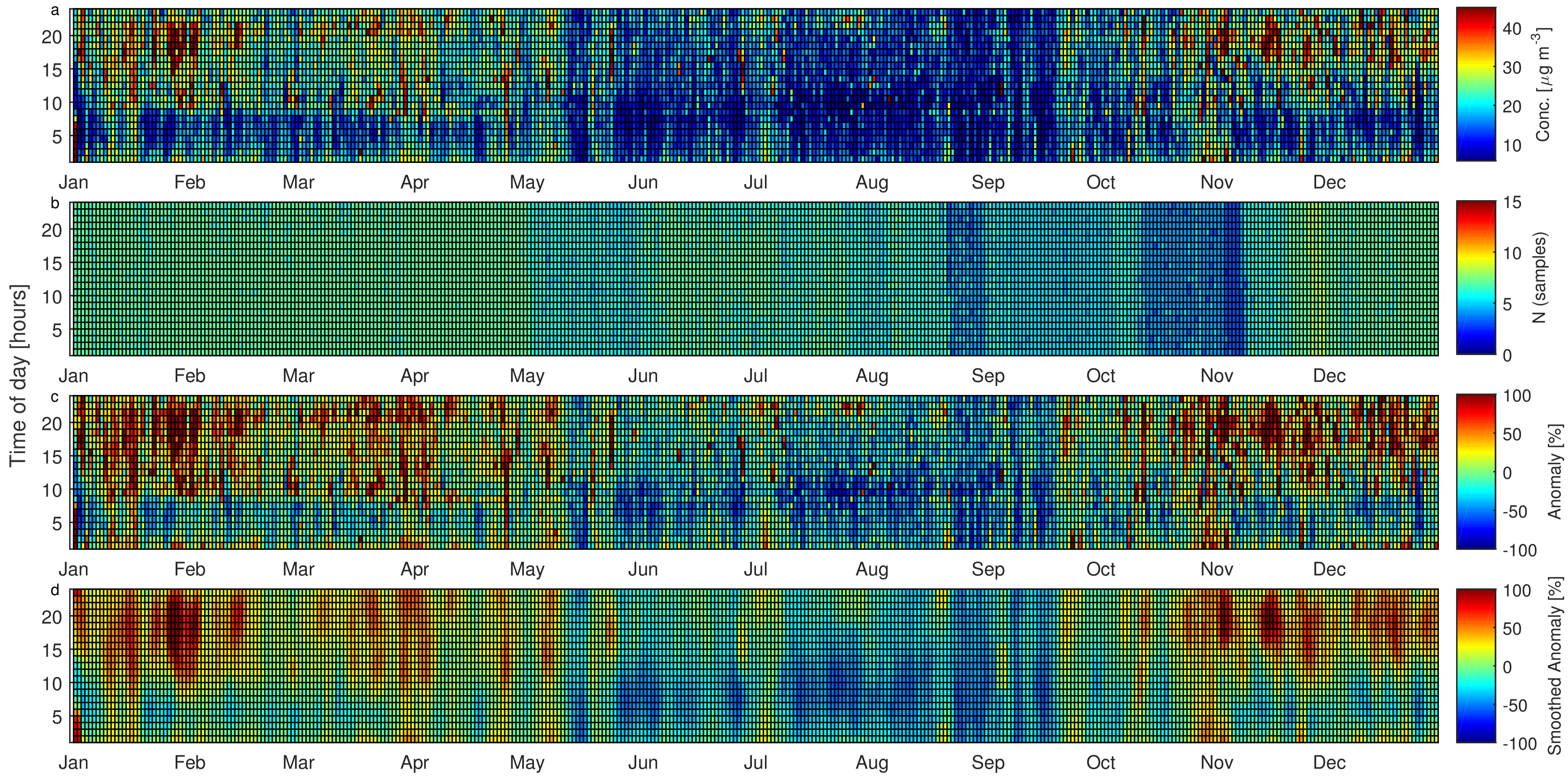

Figure 34 – PM_{10} at station *NO0063A Stener Heyerdahl*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

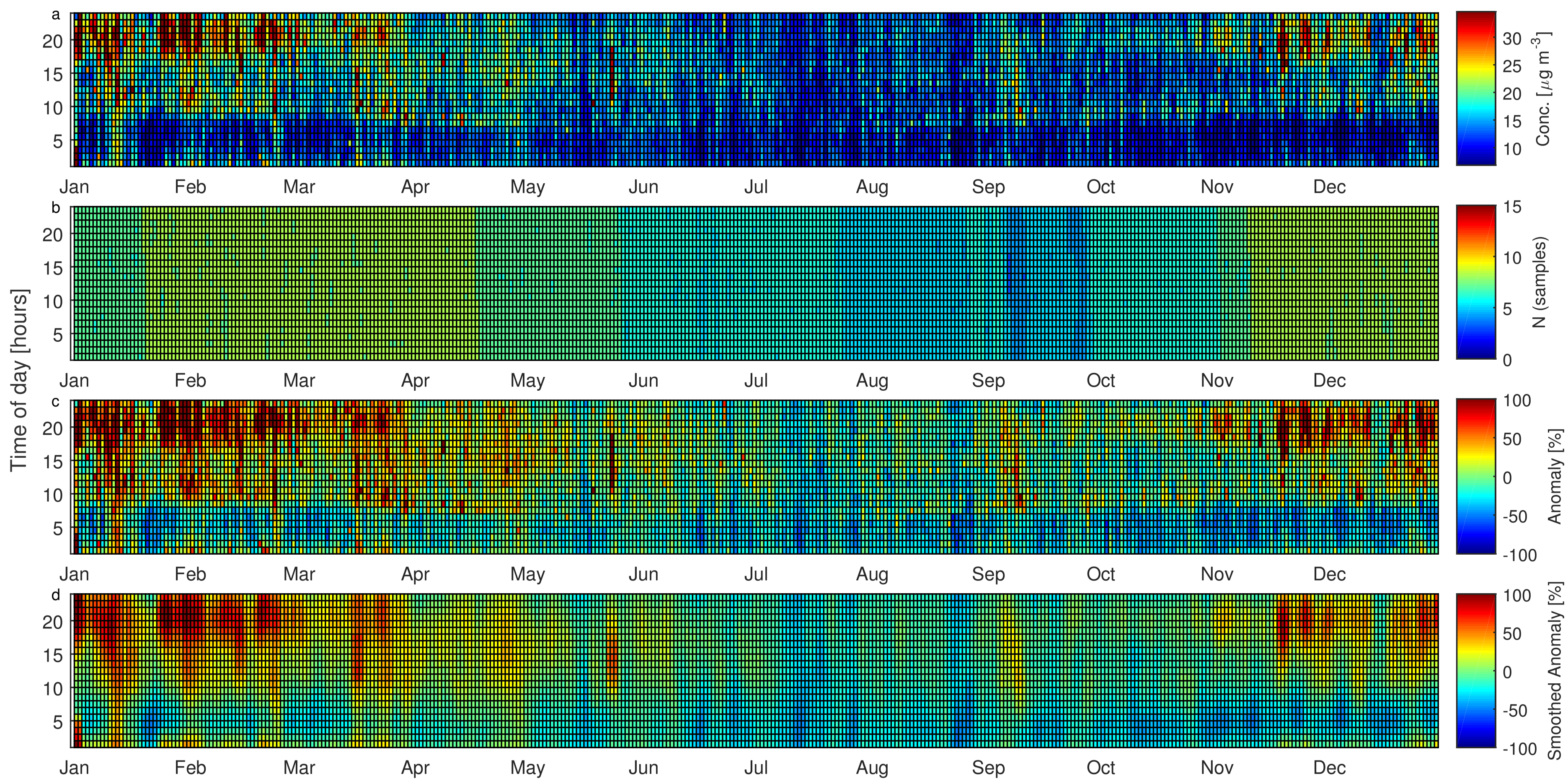

Figure 35 – PM_{10} at station *NO0065A Våland*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

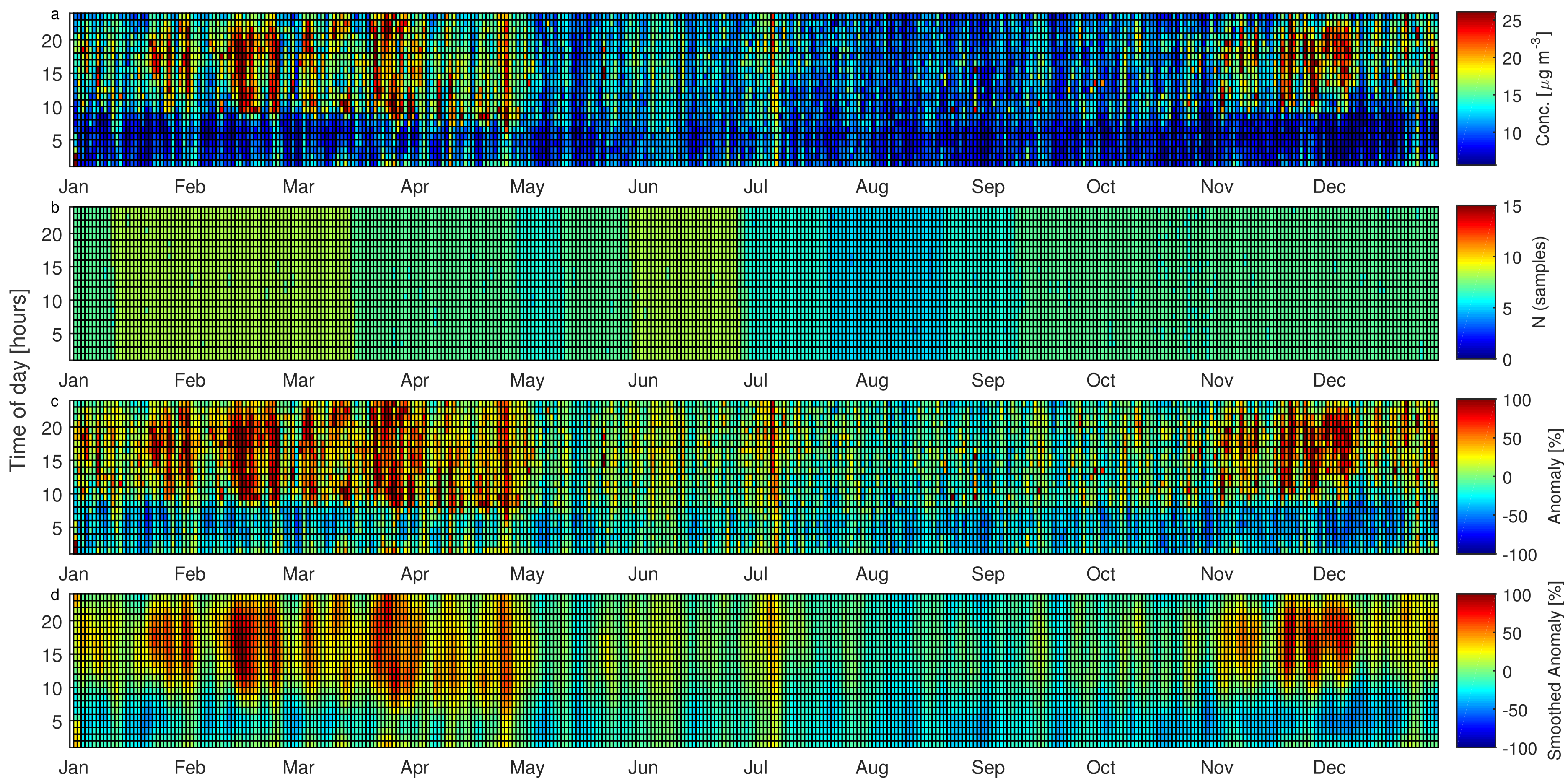

Figure 36 – PM_{10} at station *NO0070A Grimmerhaugen*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

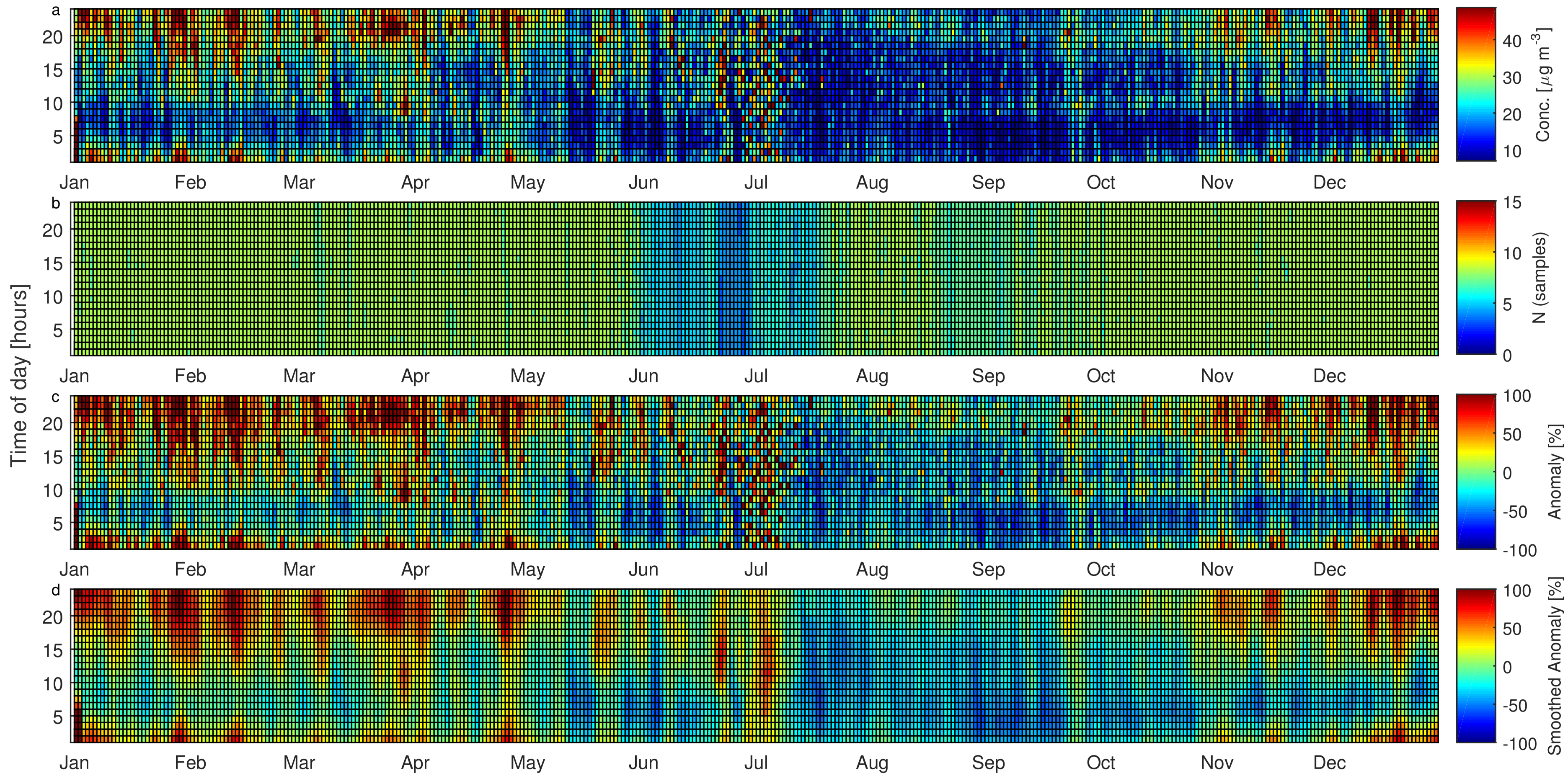

Figure 37 – PM_{10} at station *NO0072A Skøyen*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

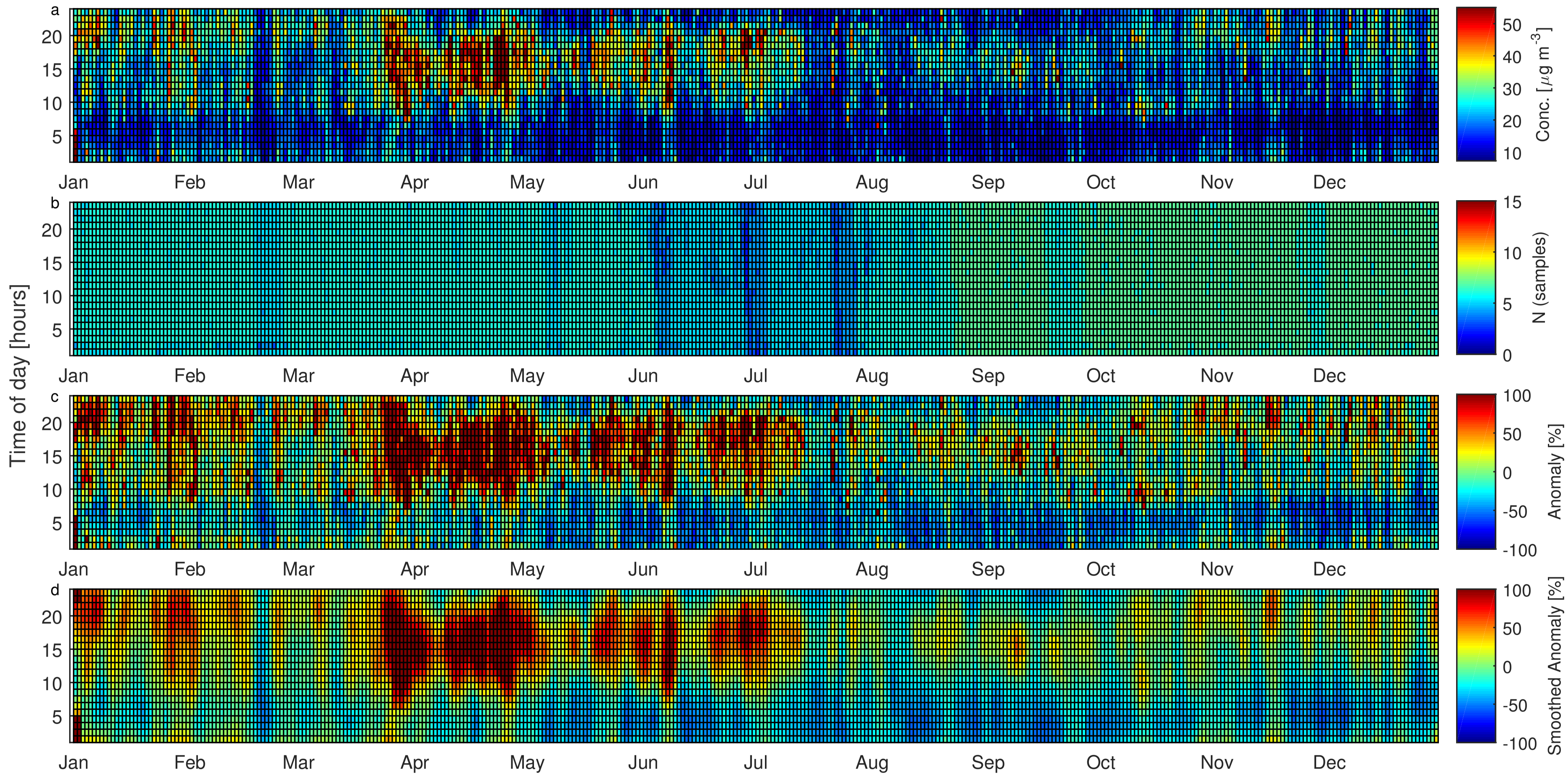

Figure 38 – PM_{10} at station *NO0073A Sofienbergparken*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

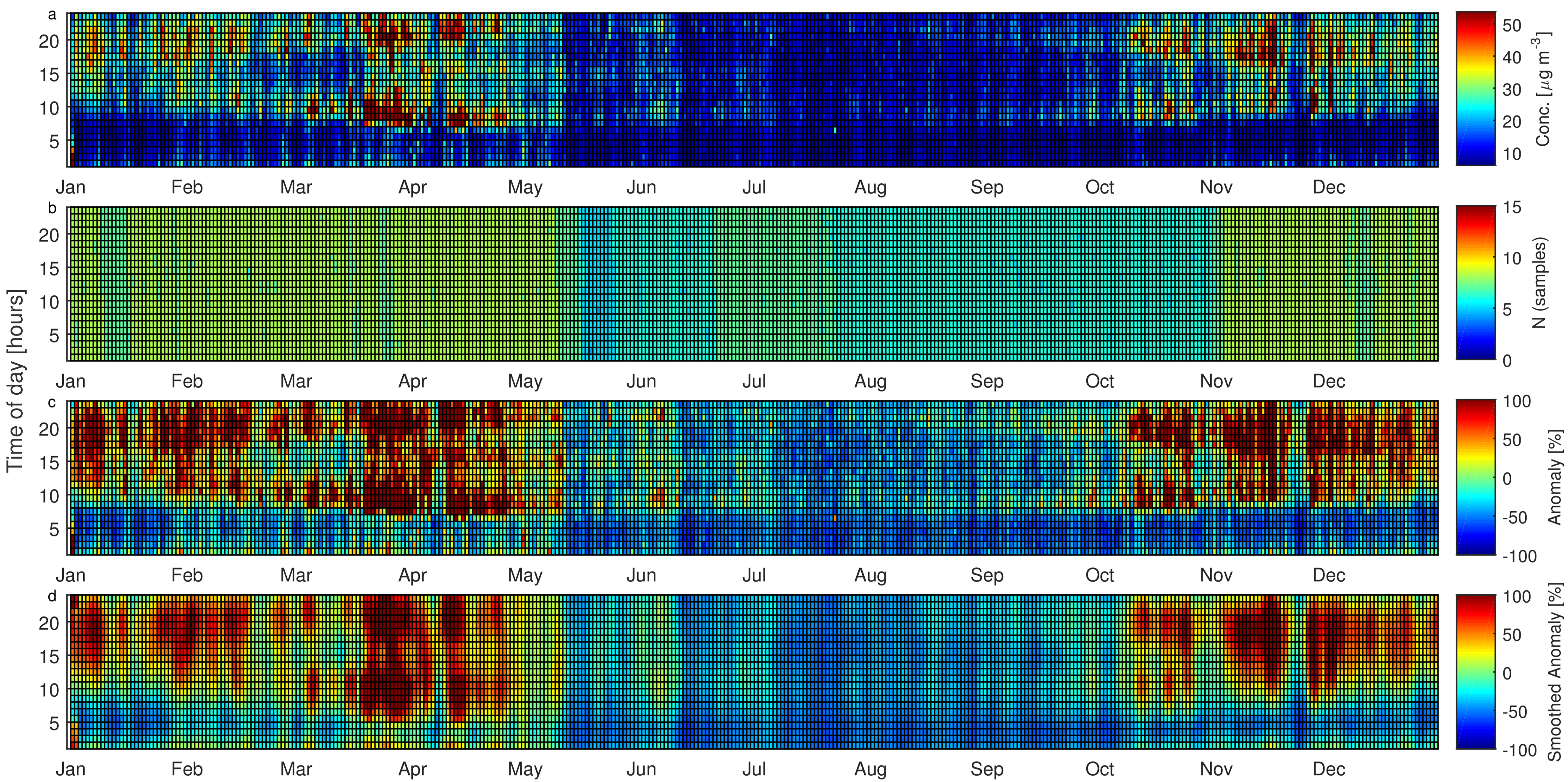

Figure 39 – PM_{10} at station *NO0075A Barnehagen*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

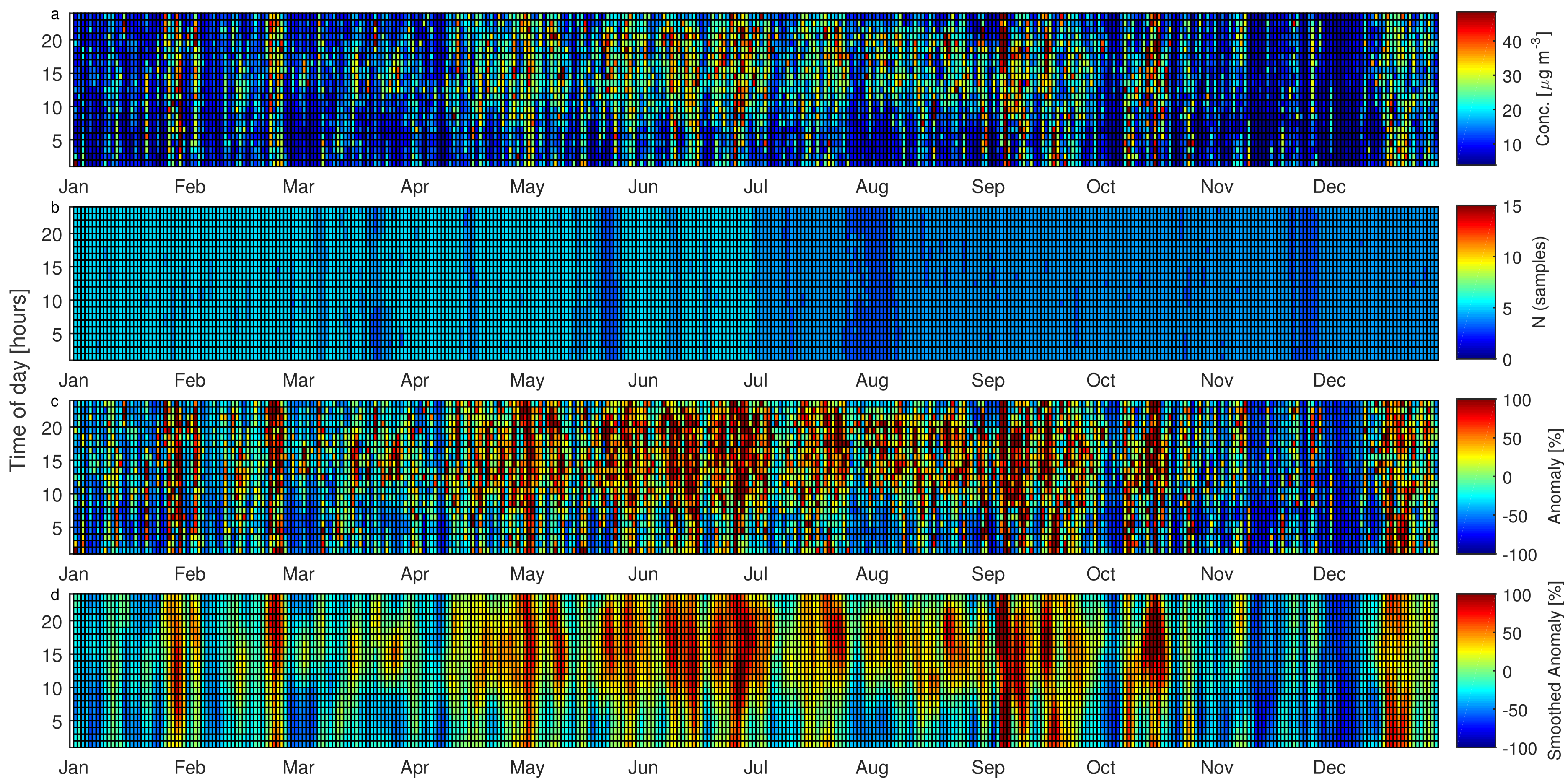

Figure 40 – PM_{10} at station *NO0077A Gruben*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

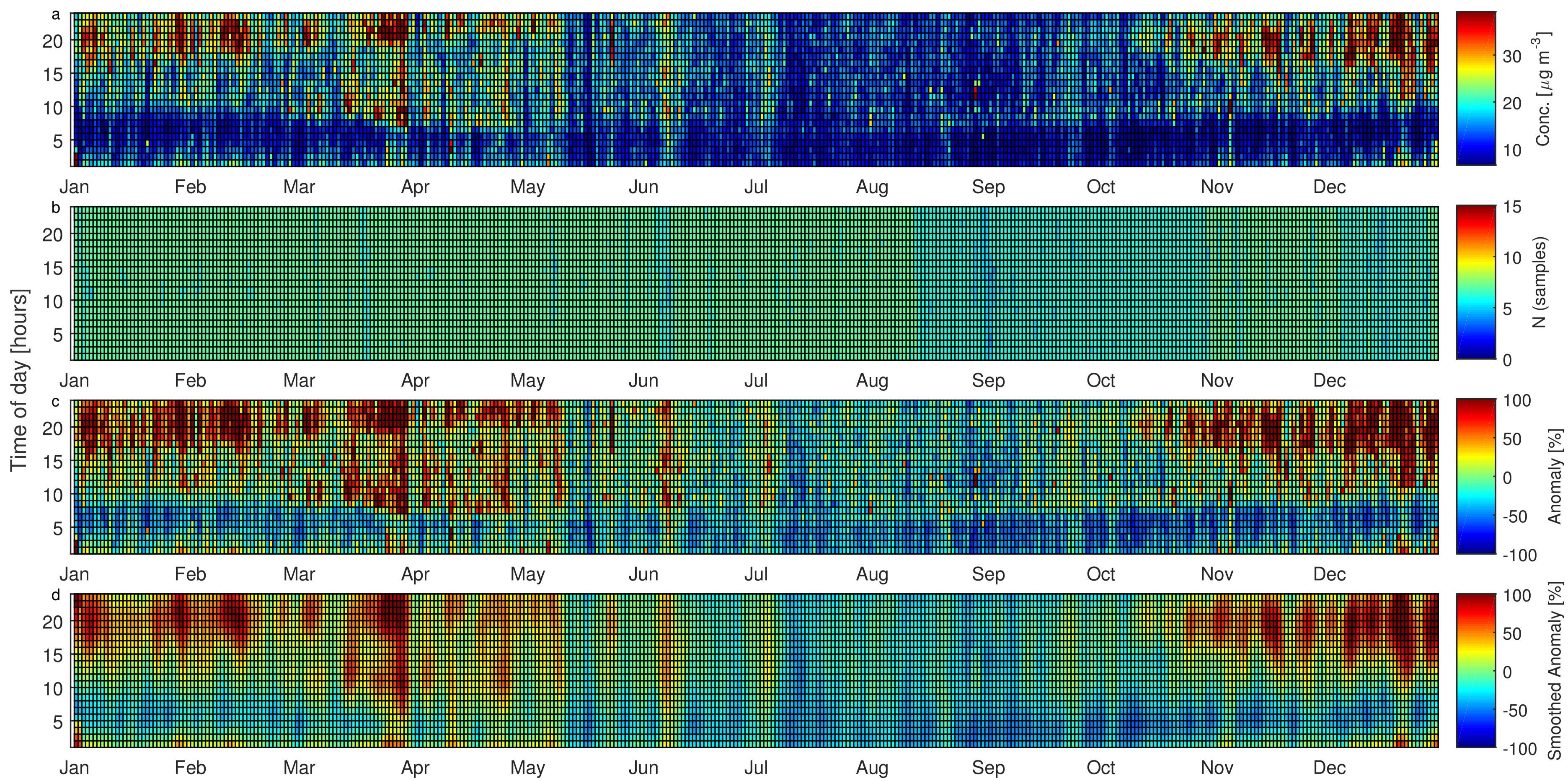

Figure 41 – PM_{10} at station *NO0080A Øyekast*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

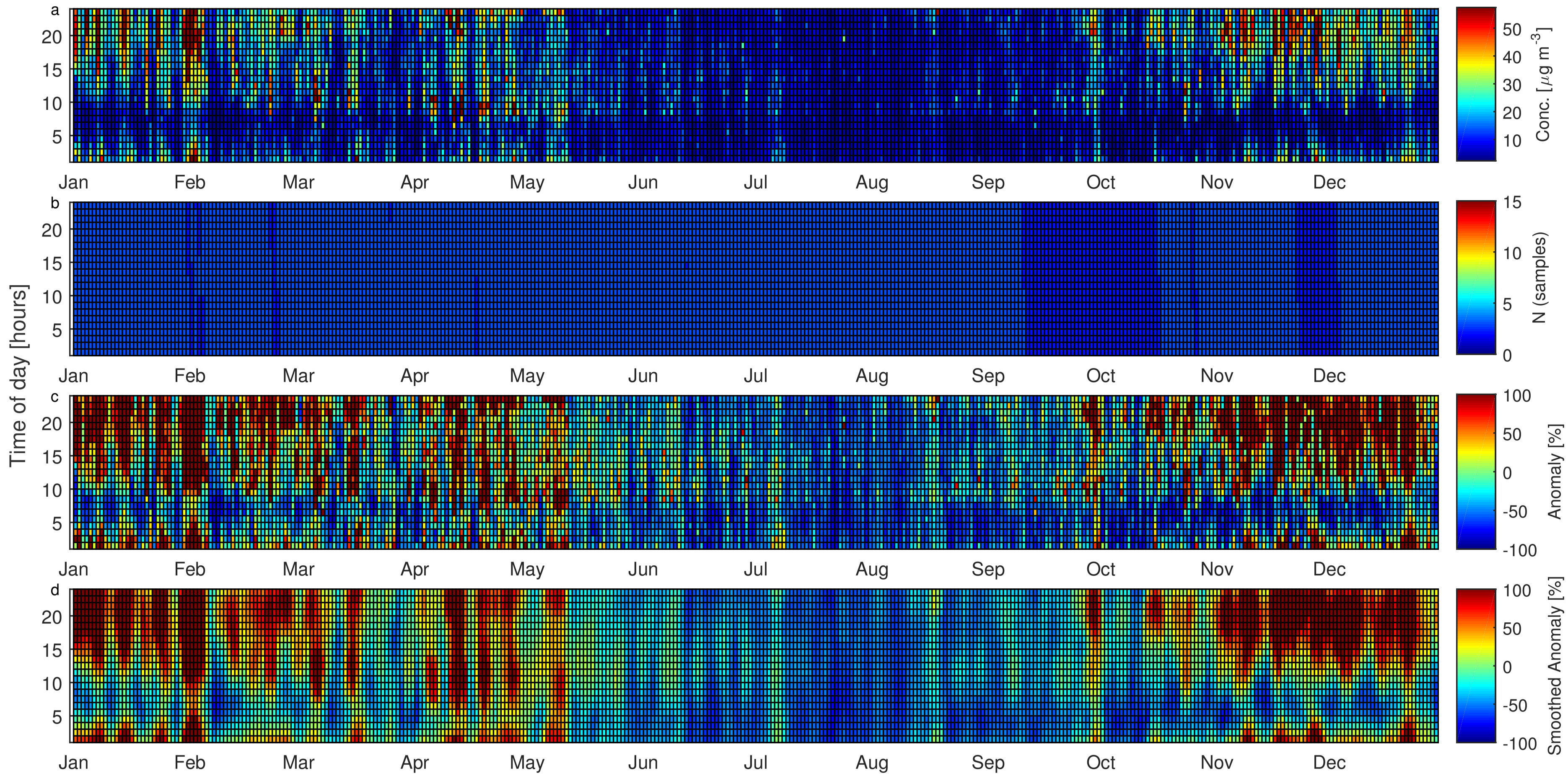

Figure 42 – PM_{10} at station *NO0089A Torvet*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

B.4 Anomaly matrices for $PM_{2.5}$

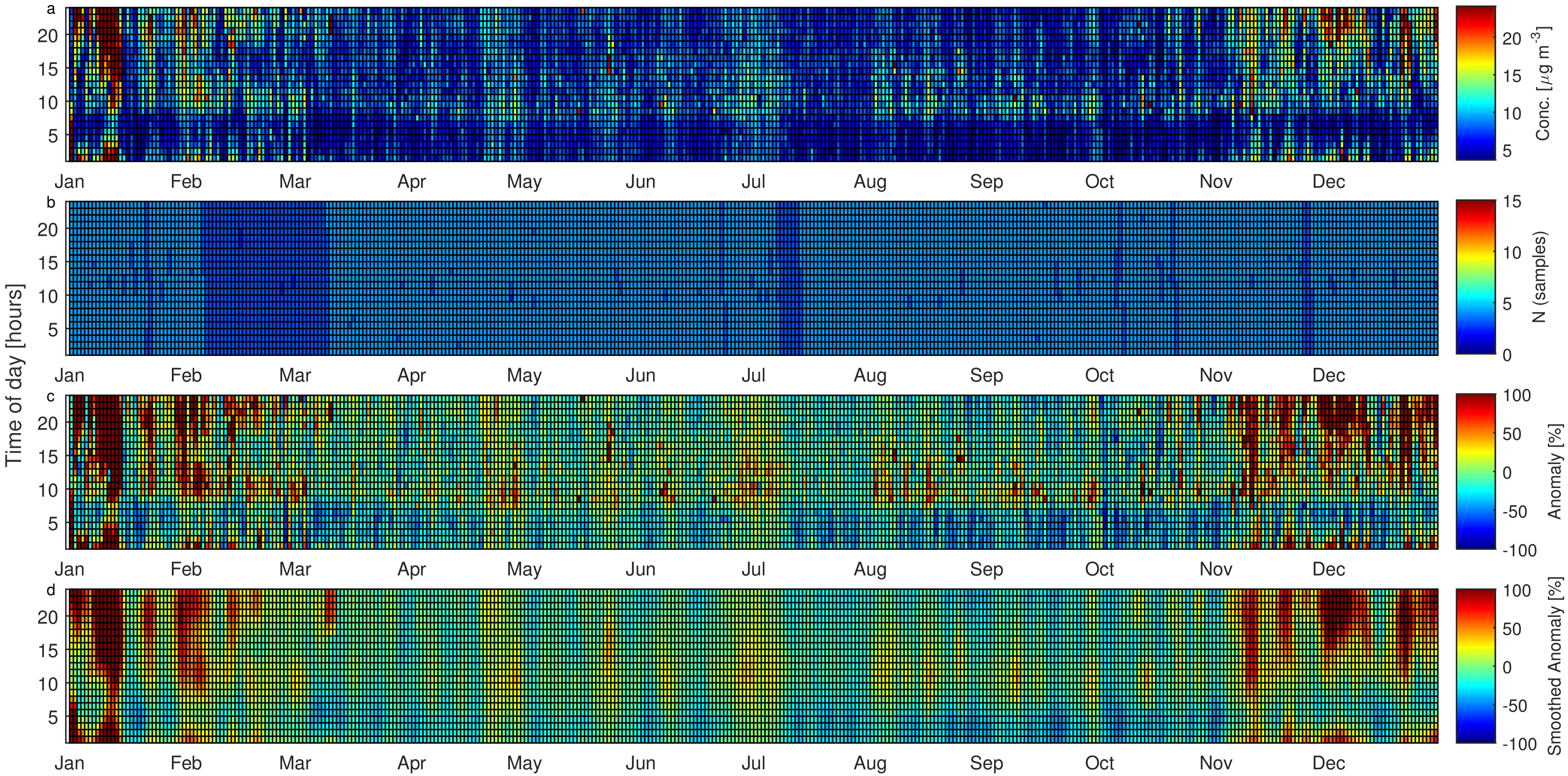

Figure 43 – $PM_{2.5}$ at station *NO0015A Rådhuset*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter. It should be noted that recent visits to this site have cast doubt on the fact if this station truly is representative of urban background. Future work should take this into account.

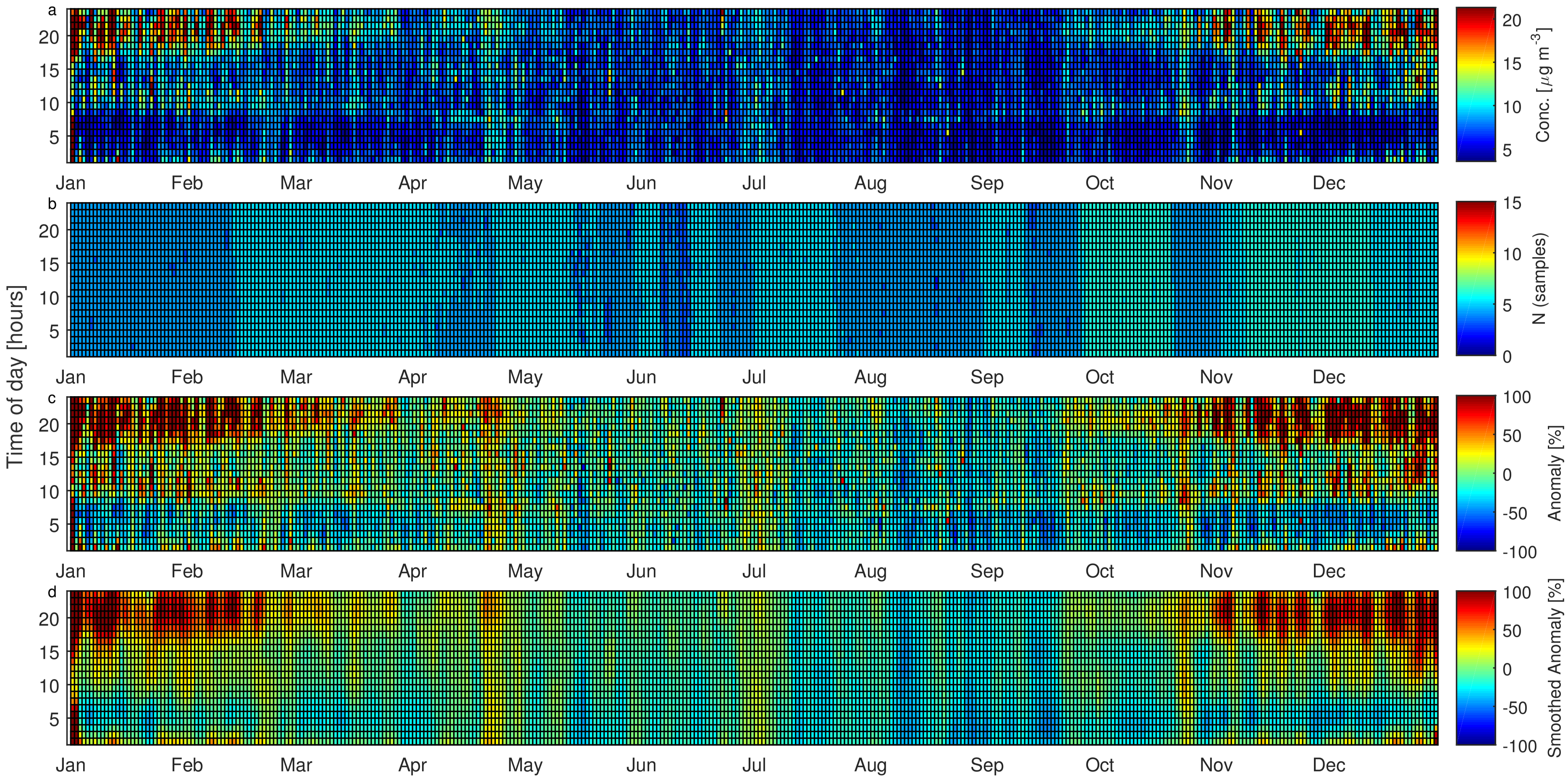

Figure 44 – $PM_{2.5}$ at station *NO0065A Våland*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

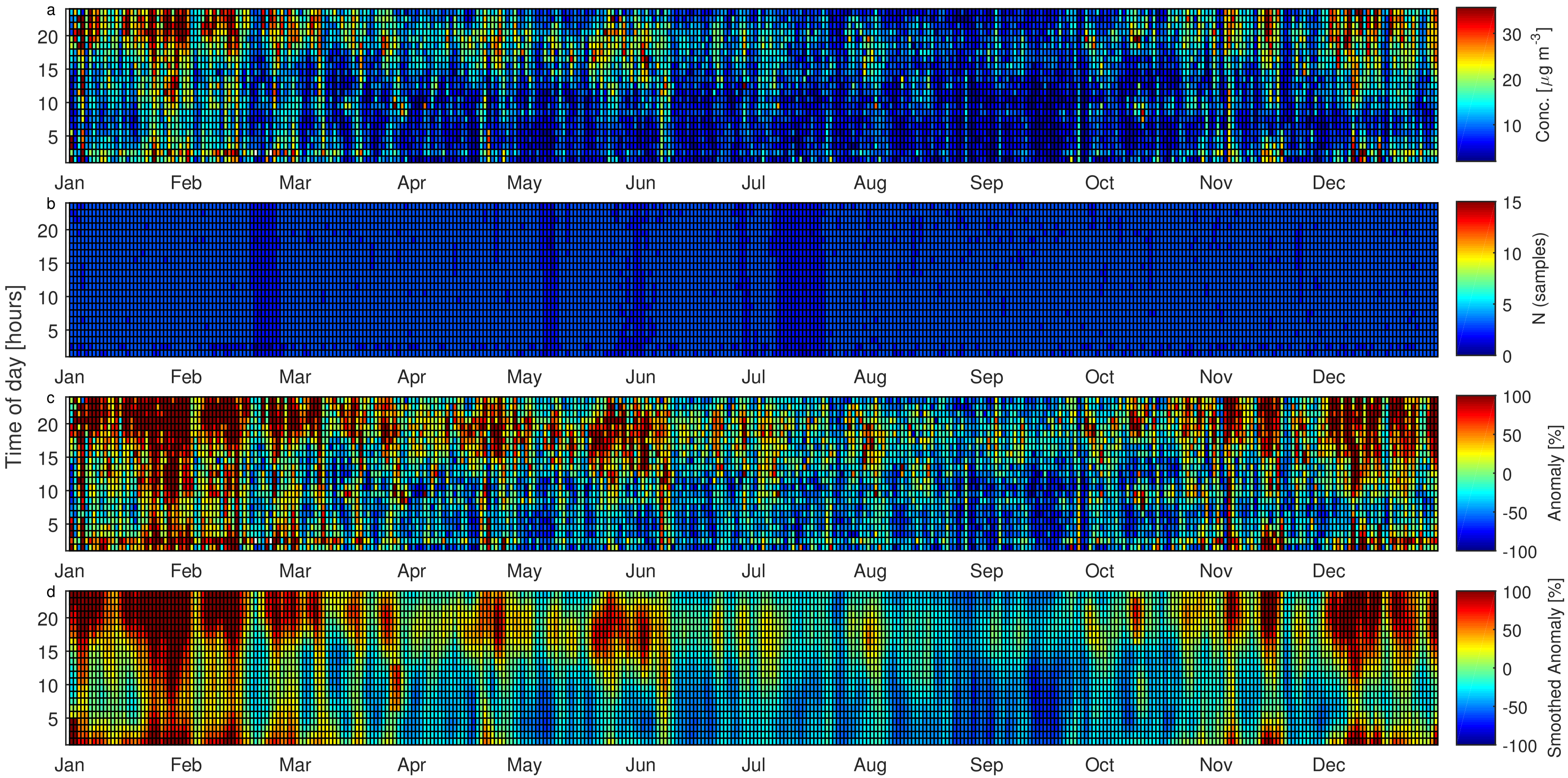

Figure 45 – $PM_{2.5}$ at station *NO0073A Sofienbergparken*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

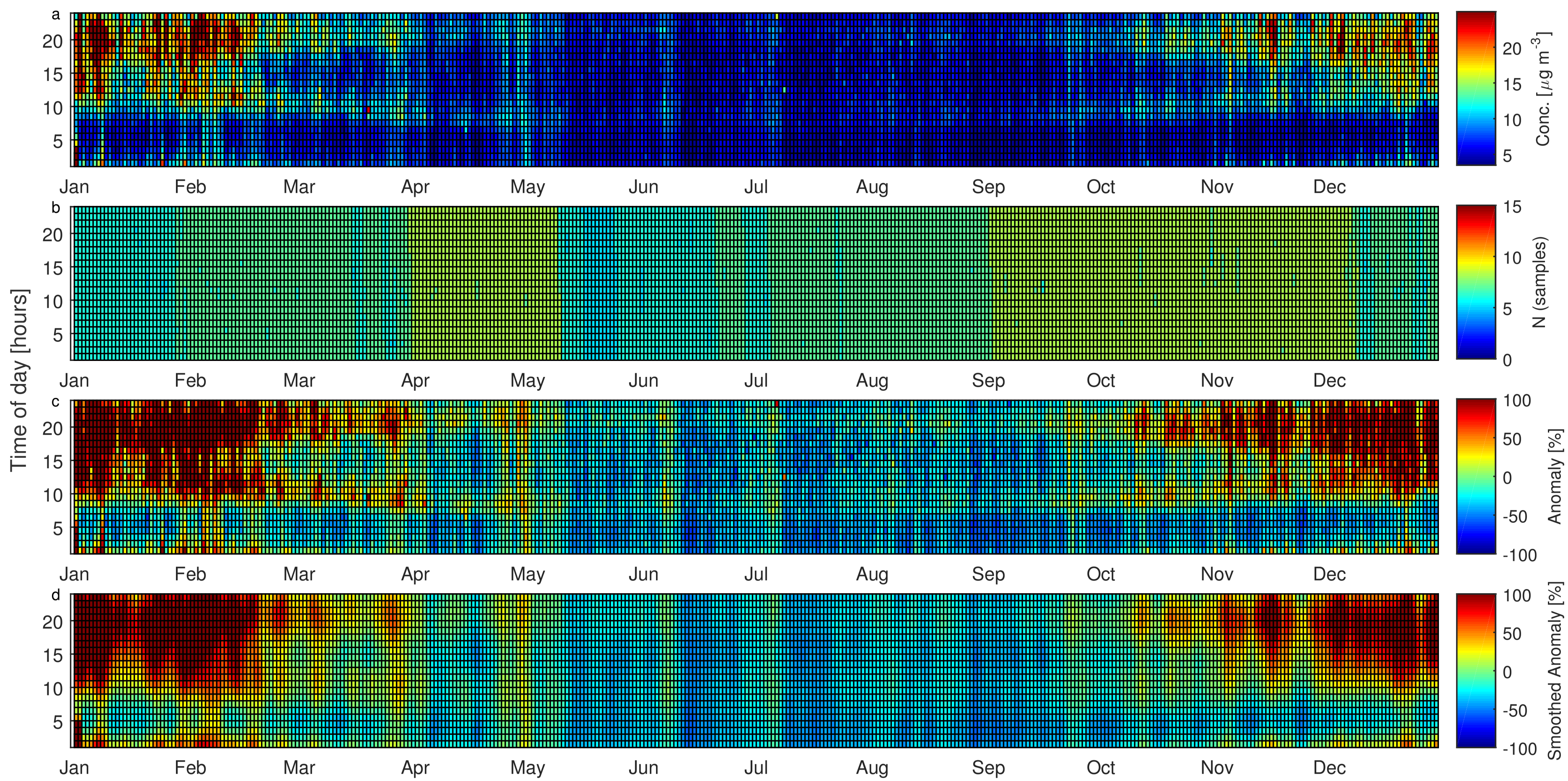

Figure 46 – $PM_{2.5}$ at station *NO0075A Barnehagen*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

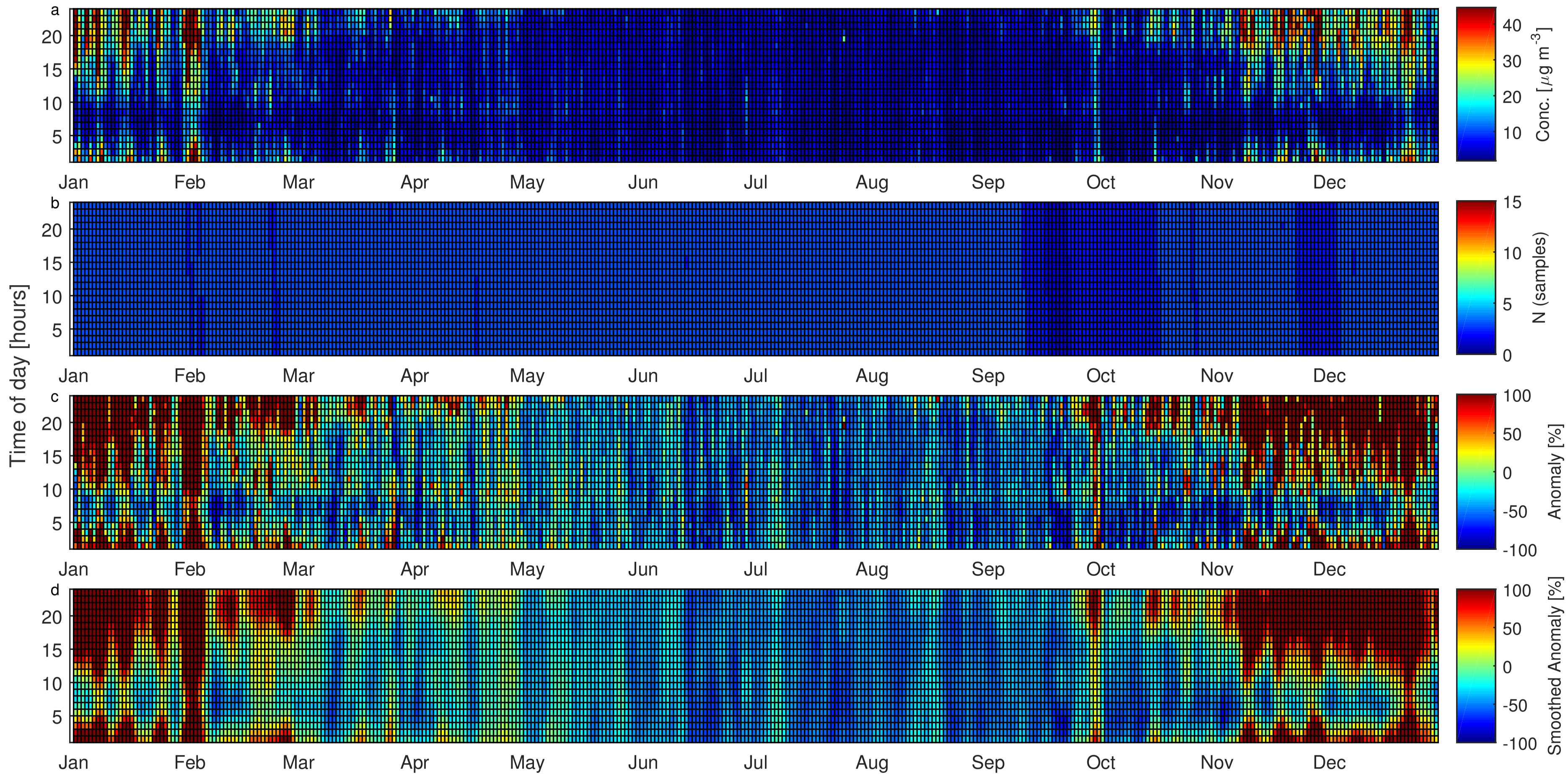

Figure 47 – $PM_{2.5}$ at station *NO0089A Torvet*: Long-term average matrices of hourly observations computed over the entire available time series, shown as a) original observations, b) number of years with available data, c) the relative anomaly computed from the long-term mean, and d) the anomaly from the long-term mean smoothed using a low-pass filter.

NILU – Norwegian Institute for Air Research
P.O. Box 100, N-2027 Kjeller, Norway
Associated with CIENS and the
Environmental Research Alliance of Norway
ISO certified according to NS-EN ISO 9001/ISO 14001

REPORT SERIES SCIENTIFIC REPORT	REPORT NO. OR 52/2014	ISBN: 978-82-425-2725-7 (print) 978-82-425-2726-4 (electronic) ISSN: 0807-7207	
DATE	SIGN.	NO. OF PAGES 86	PRICE NOK 150.-
TITLE Background concentrations in Norway - Towards automated annual updates		PROJECT LEADER Philipp Schneider NILU PROJECT NO. O-114057	
AUTHOR(S) Philipp Schneider		CLASSIFICATION * A CONTRACT REF.	
QUALITY CONTROLLER: Dag Tønnesen			
REPORT PREPARED FOR Klima- og forurensningsdirektoratet Postboks 8100 Dep 0032 OSLO			

ABSTRACT

A semi-automated technique was developed for performing annual updates of the dataset on background concentrations in Norway which was produced in previous years. The code is written in the Matlab programming language and large parts of the code base are included in the Appendix of this report.

The spatial component of the system was updated to include data from 2009 through 2011. Acquiring and preparing the input data for the spatial component still requires a relatively small amount of manual effort, however the majority of the remaining process has been automated to the largest extent possible, such that only the derivation of the semivariograms for the residual kriging step requires very brief interaction by an expert user.

The temporal component has been updated to version 8 of the European air quality database (AirBase), now including several additional years up to and including 2013. Entirely new anomaly matrices have been calculated from the updated data for all background stations in Norway.

Assuming that the availability and the format of the required input data remains unchanged, future annual updates of the system can be carried out within a very short time frame on the order of around 1-2 days.

NORWEGIAN TITLE Bakgrunnskonsentrasjoner i Norge: Automatisering av årlige oppdateringer		
KEYWORDS Air Quality	Environmental Monitoring	
ABSTRACT (in Norwegian)		

* Classification	A	Unclassified (can be ordered from NILU)
	B	Restricted distribution
	C	Classified (not to be distributed)

REFERENCE: O-114057
DATE: DECEMBER 2014
ISBN: 978-82-425-2725-7 (print)
978-82-425-2726-4 (electronic)

NILU is an independent, nonprofit institution established in 1969. Through its research NILU increases the understanding of climate change, of the composition of the atmosphere, of air quality and of hazardous substances. Based on its research, NILU markets integrated services and products within analyzing, monitoring and consulting. NILU is concerned with increasing public awareness about climate change and environmental pollution.