Climate Generator

(Stochastic Climate Representation: 120 ka to present year)

by

© Mohammad Hizbul Bahar Arif
Master of Science

A thesis submitted to the
School of Graduate Studies
in partial fulfillment of the
requirements for the degree of
Master of Science.

Scientific Computing Program
Memorial University of Newfoundland

16th December 2016

St. John's Newfoundland

Contents

Abstract **4**

Acknowledgements **5**

Abbreviations **6**

List of Figures **11**

1 Introduction **12**

1.1 Research Context 12

1.2 Downscaling 14

1.3 Bayesian inference 16

1.4 Bayesian Artificial Neural Network (BANN) 16

1.5 Markov Chain Monte Carlo integration (MCMC) 18

1.6 Gibbs sampling 18

1.7 Network architecture and prior computation 18

1.8 Automatic Relevance Determination (ARD) 19

1.9 Climate Turing Test 20

1.10 Climate Generator (CG) 20

1.10.1 Reasoning behind the name of CG 20

1.10.2 CG predictors 21
1.10.3 Climate Generator and reality 22
1.10.4 Present setup and future prospect 22
1.11 Summary 23
1.12 Thesis Overview 23

2 Stochastic climate representation for millennial scale integration over North America 24

2.1 Abstract 24
2.2 keywords 25
2.3 Introduction 25
2.4 Test study region and data 26
2.5 Methods 27
2.5.1 BANN design and training 28
2.5.2 BANN Implementation 29
2.5.3 Adding noise 31
2.6 Implementation Results 31
2.6.1 Selection of BANN architecture 32
2.6.2 Model Comparison 38
2.7 Discussion 45
2.8 Conclusions 46
2.9 Co-authorship statement 47

A Supplement of Climate Generator 51

Abstract

I present a computationally efficient stochastic climate model for large spatiotemporal scales (example, for the context of glacial cycle modelling). In analogy with a Weather Generator (WG), the model can be thought of as a Climate Generator (CG). The CG produces a synthetic climatology conditioned on various inputs. Inputs for the CG include the monthly mean sea surface temperature field from a simplified Energy Balance Model (EBM), surface elevation, surface ice, carbon dioxide, methane, orbital forcing, latitude and longitude. The CG outputs mean monthly surface temperature and precipitation using Bayesian Artificial Neural Networks (BANN) for non-linear regression. The CG is trained against the results of GCMs (FAMOUS and CCSM) over the last deglacial (22 ka to present). For validation, CG predictions are compared directly against the 120 ka to 22.05 ka interval of FAMOUS results that were not used for CG training. The stochastic noise is added to each prediction by generating the random normal distribution with mean from the ensemble networks for a single guess and Standard deviation computed from 10th and 90th percentile of the BANN predictive distribution for each time step. For the CG trained against FAMOUS, I show the predictive errors (relative to FAMOUS) are comparable to the difference between FAMOUS and the CCSM.

Acknowledgements

I would offer deep gratitude to my supervisor Lev Tarasov (Associate professor, Department of Physics and physical oceanography) for his guidance, inspiration and valuable suggestions throughout this research. Financial support was provided by MUN Scientific Computation program and RDC. Without this support, it would not have been possible to pursue this research on a full-time basis. This work has benefited greatly from the instruction provided by Tristan Hauser. Also, by assistance discussions with Taimaz Bahadory, Kevin Le Morzadec, Ryan Love and Benoit Lecavalier.

Abbreviations

ANN	Artificial Neural Network
BANN	Bayesian Artificial Neural Network
CCSM	Community Climate System Model
WG	Weather Generator
CG	Climate Generator
CCT	Climate Turing Test
CGfamous	CG train with FAMOUS climate model data
CGccsm	CG train with CCSM climate model data
EMIC	Earth systems Model of Intermediate Complexity
EOFs	Empirical Orthogonal Functions
GCM	General Circulation Model
ka	kilo year before present
MCMC	Markov Chain Monte Carlo
RMSE	Root Mean Square Error

List of Tables

2.1 Predictor sets for displayed CG results 32

2.2 Different models based on architecture and inputs 33

2.3 MEAN DIFFERENCE (MD) and RMSE relative to the FAMOUS (except for last entry) over the full grid area (train period) 33

2.4 MEAN DIFFERENCE (MD) and RMSE relative to the FAMOUS (Test period) over the full grid area 34

2.5 ARD analysis for the Model F (CGfamous) 37

2.6 Mean Difference (MD), Correlation (COR) and RMSE relative to FAMOUS (except for last entry) over the ice region (Train period) 39

2.7 Mean Difference (MD) , Correlation (COR) and RMSE relative to FAMOUS (Test period) over the ice region 39

List of Figures

2.1 Final ANN Diagram (Architecture type I) used in CGfamous and CGccsm prediction 29

2.2 ANN Diagrams are tested to improve CG prediction 32

2.3 Taylor diagram displaying a statistical comparison with FAMOUS (test part) of six different CG estimates. Pattern similarities are quantified based on their correlation and centered root mean square difference between CG and FAMOUS, and standard deviation with respect to the mean of the corresponding field. Contour grey lines indicate the root mean square (RMS) values. The model M is the same as model F but trained with CCSM climate model data. 35

2.4 Taylor diagram displaying a statistical comparison with FAMOUS (test part) of six different CG estimates. Pattern similarities are quantified based on their correlation and centered root mean square difference between CG and FAMOUS, and standard deviation with respect to the mean of the corresponding field. Contours grey line indicates the root mean square (RMS) values. The model M is the same as model F but trained with CCSM climate model data. 36

2.5 Comparison of the spatial mean (with latitudinal weighting) temperature time series. The black vertical line separates the test (left) and training part (right). 40

2.6 The August temperature field (Deg C) at 100 ka (1st and 2nd row) with elevation and ice contours shown in black and blue. The difference between plots are shown in the 3rd row. Model names are indicated in the top left corner in each box. 41

2.7 February precipitation field (cm/month). Left column: test case. Right column: Training Case. Model names and times are indicated in the top left corner in each box. 42

2.8 Temperature field, left column: Distribution of the expansion coefficient over total time steps for the leading two EOFs of the FAMOUS Model data (black), and the distribution of the expansion coefficients over time obtained by an ensemble projection of CGfamous simulation (red), EBM temperature (blue) and CCSM (cyan) onto the same EOFs. Top time series represent the August EOFs and bottom time series are the February EOFs. Right column: Distribution of the expansion coefficients for the leading two FAMOUS EOFs for the FAMOUS (black), CGfamous (red), EBM temperature (blue), and CCSM (cyan) datasets 43

2.9 Precipitation field, left column: Distribution of the expansion coefficient over total time steps for the leading two EOFs of the FAMOUS Model data (black), and the distribution of the expansion coefficients over time obtained by an ensemble projection of CGfamous simulation (red) and CCSM (cyan) onto the same EOFs. Top time series represent the August EOFs and bottom time series are the February EOFs. Right column: Distribution of the expansion coefficients for the leading two FAMOUS EOFs for the FAMOUS (black), CGfamous (red) and CCSM (cyan) datasets 44

A.1 Comparison of the spatial variance (standard deviation) on the ice region. The black vertical line separates test (left) and training part (right). 52

A.2 Comparison of the spatial mean (with latitudinal weighting) precipitation time series. The black vertical line separates test (left) and training part (right). 53

A.3 February temperature field (Deg C) at 18 ka with the elevation and ice contour shown in black and blue. Difference between plots are shown in 3rd row. Model names and months are indicated in the top left corner in each box). 54

A.4 August temperature field (Deg C) at 18 ka with the elevation and ice contour shown in black and blue. Difference between plots are shown in 3rd row. Model names and months are indicated in the top left corner in each box). 55

A.5 February precipitation field (cm/month) at 18 ka with the elevation and ice contour shown in black and blue. Models name and month are indicated in the top left corner in each box. Difference between plots are shown in 3rd row. 56

A.6 August precipitation field (cm/month) at 18 ka with the elevation and ice contour shown in black and blue. Difference between plots are shown in 3rd row. Model names and month are indicated in the top left corner in each box). 57

A.7 The August temperature field (Deg C). (1st and 2nd row) with the elevation and ice contour shown in black and blue. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box). 58

A.8 The February temperature field (Deg C) at 100 ka (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box. . . 59

A.9 The August temperature field (Deg C) at 80 ka (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box. . . 60

A.10 The February temperature field (Deg C) at 80 ka (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box. . . 61

A.11 The August temperature field at 60 ka (Deg C) (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box. . . 62

A.12 The February temperature field (Deg C) at 60 ka (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box. . . 63

A.13 February precipitation field (cm/month). Models name and times indicated in the top left corner in each box. 64

A.14 August precipitation field (cm/month) (1st and 2nd row). Difference between plots are shown in 3rd row. Model names and times are indicated in the top left corner in each box. 65

A.15 The February precipitation field (cm/month). Model names and times are indicated in the top left corner in each box. 66

A.16 The August precipitation field (cm/month). Model names and times are indicated in the top left corner in each box. 67

A.17 February correlation map between residual (i.e. CG without noise - FAMOUS) Temperature and precipitation fields. 68
A.18 August correlation map between residual (i.e. CG without noise - FAMOUS) Temperature and precipitation fields. 69

Chapter 1

Introduction

1.1 Research Context

Computational cost is a key issue for glacial cycle modelling, particularly for paleoclimate modelling. For large spatio-temporal time-scales even previous generation GCMs are prohibitively expensive, involving millions of cpu-hours for a single simulation. The statistical correction of faster simplified climate models to better approximate the predictive quality of GCMs constitutes the central theme of this thesis. From a Bayesian framework, I extract a posterior for climate conditioned on various inputs, including the output of a fast 2D Energy Balance Model (EBM). Natural variability or climate noise are imposed in model prediction to each time steps through the addition of gaussian noise, with noise variance extracted from the afore mentioned posterior. I focus on constructing an efficient stochastic climate representation to provide a better climate representation for glacial cycle (120 ka to present year) modelling. Towards this goal, the small scale Weather Generator (WG) concept is implemented on a large spatiotemporal scale, and accordingly, is named a "Climate Generator (CG)".

General Circulation Models (GCMs) are an established tool for estimating the large scale evolution of the Earth's climate. They represent the physical processes occurring in the atmosphere, ocean, cryosphere and land surface and their interactions. In GCMs, core mathematical equations that are derived from mathematical laws (conservation of energy, conservation of mass, conservation of momentum and the ideal gas law) are solved numerically. These models produce a three-dimensional

picture of the time evolution of the state of the whole climate system. Current GCMs are too computationally expensive to run continuously over O(100 kyr) glacial cycle time scales. For example, the simplified low resolution (atmospheric part of the model has resolution $5\,^{\circ} \times 7.5\,^{\circ}$) FAMOUS GCM has been run for the entire last glacial cycle (LGC period, 120 ka to present year) only in an highly accelerated mode. This model can run at the rate of 250 years in a day on eight cores [Smith and Gregory, 2012]. The longest integration of a full complexity GCM to date took 2 years to complete the 22 ka to present deglacial interval (Community Climate System Model (CCSM), with a T31 atmospheric component [Liu et al., 2009]). As a fast alternative, Energy Balance Models (EBMs) can integrate a whole glacial cycle in a day or less. They predict the surface temperature as a function of the Earth's energy balance with diffusive horizontal heat transports. However, in an EBM, atmospheric dynamics are not modelled and only the sea level temperature field is computed on the basis of energy conservation. Given this, the resolution of the EBM is kept low (T11 @ 1500 km) and has no precipitation field [Hyde et al., 1990], [Tarasov and Peltier, 1997].

Coupled icesheet and climate modelling over a full glacial cycle is an example where computational speeds currently preclude the use of GCM climate representations, especially for large ensemble-based analyses required for assessing dynamical and/or reconstructive uncertainties. Earth System Models of Intermediate Complexity (EMICSs) enable long-term climate simulations over several thousands of years but are at the edge of applicability for a full glacial cycle. For example, LOVECLIM is a low resolution (Atmospheric component is T21) climate model. It takes about 15 days to run 10 kyr, [Goosse et al., 2010]). Thus, there remains a need for a faster climate representation (temperature, precipitation etc.) for last glacial cycle (120 ka to present year) ice sheet modelling, especially in large ensemble contexts.

To do this, a new approach is proposed for efficient climate modelling over large spatio-temporal scales: the Climate Generator (CG). The CG uses the results of previous GCM runs to effectively improve the output of a fast simplified climate model (in this case an EBM) and thereby provide a stochastic representation of climate that runs approximately at the speed of the fast model. The Climate Generator (CG) can also be understood as a field-specific emulator for GCMs. This is because we train our CG using GCM data to make climate predictions without the computational expense of running a full GCM. As an alternative view, the CG operates similar to aspects of downscaling tools. Downscalling tools are generally used to increase resolution in certain climate characteristics. Similarly, the CG is developed based on mainly coarse resolution climate representations and

converts EBM temperature to a GCM scale. The CG produces temperature and precipitation fields using a EBM temperature field as input, similar to downscaling techniques for temperature.

1.2 Downscaling

Most GCMs neither assimilate nor provide information on scales smaller than a few hundred kilometers for the atmosphere. The relevant scale of climatic impacts is much smaller than this. We have, therefore, a spatiotemporal downscaling problem (Fowler and Wilby [2007]). Downscaling methods create a relationship between the state of some variables on a large scale and the state of the variables on a smaller scale. These methods are used to convert the GCM output to a local scale. The two main approaches to downscaling are Dynamical Downscaling (DD) and Statistical Downscaling (SD). Both methods can provide a high-resolution regional climate, but DD is computationally expensive and requires large amounts of GCM boundary forcing data Fowler and Wilby [2007]. SD needs a long and reliable observed historical data series for calibration and depends on the choice of predictors, domain size, and climate region. But SD approaches are simple to implement and computationally inexpensive.
To create such a SD tool, the following conditions must be fulfilled (Schoof [2013], [Benestad et al., 2008]):

- A strong relationship between large-scale predictor and small-scale predictand.
- Predictors are simulated well by the models.
- Statistical relationship between the predictor and predictand that does not change over time.

SD methods have been categorized (based on application technique) as regression-based methods (Multivariate Regression (MVR), Singular Value Decomposition (SVD), Canonical Correlation Analysis (CCA), Artificial Neural Network (ANN)), weather pattern based method (fuzzy classification, self-organizing map, Monte Carlo methods) and weather generators (Markov chains, stochastic models, Schoof [2013]). Regression-based methods are relatively straightforward to apply and simple to handle but have an inadequate representation of observed variance and extreme events [Wilby et al., 2004]. CCA finds spatially coherent patterns in various data fields that have the largest possible correlation, whereas SVD finds coupled spatial patterns that have maximum temporal covariance [Benestad et al., 2008]. MVR optimizes the fit (minimizing the RMSE). Regression-based methods

are widely used in hydrological response assessment [Chu et al., 2010]. Weather-pattern based methods are often used in the analysis of extreme events. Weather Generators (WGs) replicate the statistical attributes of local climate variables rather than the observed sequences of events [Wilby et al., 2004]. Regression and weather pattern based methods have been jointly implemented via ANNs (e.g. combined principle components of multiple circulations as predictors in an ANN for winter precipitation) [Schoof, 2013]. The statistical downscaling method (SDSM) is a hybrid of a regression method and weather generator [Chu et al., 2010]. Statistical methods are chosen based on the nature of local predicted variables. A relatively smooth variable, such as monthly mean temperature, can be reasonably represented by regression-based methods. If the local variable is highly discontinuous in space and time, such as daily precipitation, it will require a more complex non-linear approach [Benestad et al., 2008].

Artificial Neural Networks (ANNs) are simple to implement and are non-parametric. As a result, ANNs are widely used for modelling the complex relationship between inputs and outputs in a short spatial or temporal scale climate prediction, such as forecasting problems [Reusch and Alley, 2004]. If the input and output relationships are nonlinear or if the output is non-Gaussian, ANNs give better fits to observations than standard parametric approaches e.g. for the case of daily precipitation [Schoof and Pryor, 2001], [Dibike and Coulibaly, 2006]. However, ANNs do not provide uncertainty estimates, and over-fitting is the most common cause of poor predictive ability. Bayesian Artificial Neural Networks (BANNs) address these deficiencies. BANN is a non-linear regression-based method where Bayesian inference is implemented on ANNs. BANN have been used to create weather generators [Hauser and Demirov, 2013], perform climate model calibration [Hauser et al., 2012] and make short scale climate predictions [Lu and Qin, 2014]. I have chosen BANNs as an emulator to construct my CG for the following reasons:

- BANNs are ensembles of ANNs drawn from a probability distribution derived by training the network against available data using Bayesian inference.
- The training procedure of BANNs estimates network parameters based on the fitted parametric noise model.
- BANNs provide a prediction consisting of an expected value together with an associated uncertainty Lee [2007]).
- Over-fitting is generally avoided in BANNs because their learning procedure is based on

minimizing the sum of the bias and the variance squared.

- There is no need for separate cross validation on data sets, so that the whole data set can be used for training and validation.
- The amount of training data is irrelevant in BANNs for adjusting the complexity of the model because, in the Bayesian perspective, a posterior distribution is not going to maximize fit (unless the prior is uniform). In other terms, the prior regularizes the fitting.
- Automatic Relevance Determination (ARD) can identify which inputs are most relevant to predict the target. This is important because including inputs of low statistical relevance will likely degrade predictive performance Neal [2012].

The CG uses a BANN for non linear regression.

1.3 Bayesian inference

In Bayesian inference, Bayes theorem is applied to derive the posterior probability, θ, given the data, D,

$$P(\theta/D) = \frac{P(D/\theta)P(\theta)}{P(D)} \propto L(\theta|D)P(\theta), \tag{1.1}$$

where P(θ) is the prior distribution for the parameters, (that express our initial beliefs about their values, before any data has arrived) and P(D/θ) is called the likelihood. A prediction of an unknown quantity C is given by the expection of the C relative to the posterior distribution of the parameters, sic

$$P(C|D) = \int P(C|\theta)P(\theta|D)d\theta. \tag{1.2}$$

Bayesian inference implemented on ANNs is known as Bayesian Artificial Neural Networks (BANNs). The BANNs procedure is briefly described in Neal [2012]. The following sections 1.4, 1.5 and 1.7 are presented a summary of the BANNs procedure based on Neal [2012].

1.4 Bayesian Artificial Neural Network (BANN)

In BANNs, Bayesian inference is applied to ANN to generate a posterior probability distribution for the network weights, W, given the training data, D. In an ordinary ANN (with one hidden layer),

the output is computed as follows:

$$f_k(x) = b_k + \sum_j v_{jk} h_j(x) h_j(x) = tanh(a_j + \sum_i u_{ij} x_i), \tag{1.3}$$

where i is the input unit, j is the hidden unit and k is the output unit. The u_{ij} and v_{jk} are the weights on the connection between i to j and j to k respectively. Biases of the hidden and output units are given by a_j and b_k. These weights and biases are known as the network parameters. Each $f_k(x)$ (output value of the network) is weighted by the sum of hidden values and a bias. The value of each hidden unit $h_j(x)$, is computed with a non-linear activation function, in our case the hyperbolic tangent (tanh). In the case of a regression model (real value targets), we have for distinct outputs given the input:

$$P(y|x) = \prod_k \frac{1}{\sqrt{2\phi}\sigma_k} exp(\frac{-(f_k(x) - y_k)^2}{2\sigma_k^2}). \tag{1.4}$$

Where the targets y_k and input x are in this case modelled by Gaussian distribution with mean $f_k(x)$ (corresponding network outputs) and a standard deviation given by the hyperparameters σ_k (noise levels for the targets).

Consider the set of training cases, $(x^1, y^1), (x^2, y^2), ..., (x^n, y^n)$ and replacing: $C = x^{n+1}$ (predictive distribution) and $D = (x^1, y^1), (x^2, y^2), ..., (x^n, y^n)$; the equation 1.2 is modified as follows (for the target values in new test case):

$$P(y^{n+1}|x^{n+1}, (x^1, y^1), (x^2, y^2), ..., (x^n, y^n)) = \int P(y^{n+1}|x^{n+1}, \theta) P(\theta|(x^1, y^1), (x^2, y^2), ..., (x^n, y^n)) d\theta. \tag{1.5}$$

Here, θ, stands for the network parameters (weights and biases) and the likelihood (defined in equation 1.1) is modified as:

$$L(\theta|(x^1, y^1), (x^2, y^2), ..., (x^n, y^n)) = \prod_{i=1}^{n} P(y^i|x^i, \theta). \tag{1.6}$$

The component of y^{n+1} is estimated to be the mean of its predictive distribution and defined as:

$$y_k^{n+1} = \int f_k(x^{n+1}, \theta) P(\theta|(x^1, y^1), (x^2, y^2), ..., (x^n, y^n)) d\theta. \tag{1.7}$$

where f_k, the network output functions, are relying on the network parameters θ. A computationally efficient Markov Chain Monte Carlo (MCMC) method is used to compute the above integration.

1.5 Markov Chain Monte Carlo integration (MCMC)

Bayesian prediction may take the form as in equation 1.5 or of a single-valued prediction as in equation 1.7. In both cases, the expectation value of the function is computed relative to the posterior probability density for the parameter (say A(θ)). The expectation value of a continuous random variable a(θ) is defined as:

$$E[a] = \int a(\theta)A(\theta)d\theta. \tag{1.8}$$

Expectation values defined in equation 1.8 are approximated by the Monte Carlo method, taking a sample from A:

$$E[a] \approx \frac{1}{N}\sum_{t=1}^{N} a(\theta^t), \tag{1.9}$$

The Monte Carlo integration formula of equation 1.9 gives an unbiased estimate of $E(a)$ and converges to an accurate value with increasing N.

1.6 Gibbs sampling

Any statistic can be computed from a posterior distribution, by using equation 1.9, as long as N simulated samples from that distribution is available. To do this task, Gibbs sampling is one MCMC technique, which is use to obtain a sample from the posterior distribution. This technique is applicable when sampling of one parameter is computed at a time from a distribution conditioned on all the other parameters. This is an iterative algorithm, where the random sample of one variable is drawn at a time from its conditional distribution with the remaining variables fixed to their last updated values. BANN hyperparameters are determined via Gibbs sampling (as the conditional distribution for a single hyperparameter is accessible). A Hybrid MCMC algorithm is applied to sample the weights and biases of the BANN.

1.7 Network architecture and prior computation

An ordinary network has zero or more hidden layers. The first hidden layer is joined to the inputs, and the rest of the layers are linked to the previous hidden layer, and optionally to the input. The output units are connected to the last hidden layer (and at times to other hidden layers or input

units). The prior distribution for the parameters of a network is defined in terms of hyperparameters, that control the standard deviation for weights and biases in different groups. Neal [2012] defines these hyperparameters in terms of "sigma values (σ)" and their distribution are assigned in terms of the respective "precision", $\tau = \sigma^{-2}$, which are sampled from Gamma distributions. Consider u_1, u_2, ..., u_k as the parameters in one group. The hyperparameter for this group given the standard deviations (σ_u) of a Gaussian prior is defined as:

$$P(u_1, u_2, ..., u_k|\sigma_u) = 2(\pi)^{-k/2}{\sigma_u}^k exp(-\sum {u_i}^2/2{\sigma_u}^2). \tag{1.10}$$

τ_u is the precision and is defined as $\tau_u = {\sigma_u}^{-2}$.

The prior for the hyperparameter is expressed as (with mean w_u):

$$P(\tau_u) = \frac{(\alpha_u/2w_u)^{\alpha_u/2}}{\Gamma(\alpha_u/2)}(\tau_u)^{\alpha_u/2-1}exp(-\tau_u\alpha_u/2w_u). \tag{1.11}$$

The prior for τ_u is controlled by the values of α_u (positive). The process of computing the prior is hierarchical with a three-level approach. At the base level, each parameter is assigned a Gaussian distribution with zero mean associated with some precision. At the following level, the precision is sampled from a Gamma distribution with a particular shape parameter and with a mean given by a hyperparameter (common to all parameters of the same subgroup). The high-level hyperparameter is chosen from a Gamma distribution with a specified mean and with a specified shape parameter. When training data is obtained, the prior is updated to a posterior parameter distribution and is then used to make predictions for a test case. A similar three-level approach is used to fit the noise levels.

1.8 Automatic Relevance Determination (ARD)

The number of predictors used in modelling the distribution of a predictand is a source of complexity in ANNs. Including more and more inputs leads to poor predictive performance since irrelevant inputs will, by chance, appear in the finite training set to be more closely associated with the targets than are the truly relevant inputs. So the number of input variables must be limited, based on estimation of which attributes are most likely to be relevant. To accomplish this in BANNs, individual hyper-parameters weight the contribution of inputs to the network to maximize predictive performance. Neal [2012] describes this process as "Automatic Relevance Detection" (ARD). In

ARD, each input variable has an associated hyperparameter that controls the magnitude of the weight on the connection out of that input unit. If the hyperparameter that is associated with an input identifies a small standard deviation (around zero) for weight out of that input, these weights will likely be small, and the input will have little effect on the output, whereas, the significance of the input will increase if the hyperparameter specifies a large standard deviation.

1.9 Climate Turing Test

In the research of artificial intelligence, the Turing Test is performed to determine if a machine's ability to exhibit intelligent behaviour is indistinguishable from that of a human. The machine will pass the test if it is intelligent and responsive in a manner that is indistinguishable from a human. In this project, the Turing test concept is implemented as a Climate Turing test, to ascertain if the pre glacial maximum behaviour (120 ka to 22.05 ka) of climate predicted by our CG is indistinguishable from the FAMOUS climate model. To execute the Climate Turing test, the difference between CCSM and FAMOUS is taken as a reference uncertainty to compare the difference between the CG and FAMOUS. The comparisons are made based on the mean correlation (over space and time), mean deviation, Root Mean Square Error (RMSE), map plots and projections of Empirical Orthogonal Function (EOF) of our CG predictions with FAMOUS climatological simulated fields (monthly mean temperature and precipitation). To pass the Climate Turing test, simulated fields should have relatively high correlation (i.e. with respect to GCM target field), small RMSE, close patterns and a reasonable capacity to capture natural variability compared to the reference uncertainty.

1.10 Climate Generator (CG)

1.10.1 Reasoning behind the name of CG

Weather Generators (WGs) are computationally useful statistical tools that can be used to generate realistic and rapid daily sequences of atmospheric variables, such as temperature and precipitation, on a small scale. WGs are a means of generating a random time series of 'weather' that replicates observed statistics. It can be used to investigate small-scale climate impacts and to compute numerous random realizations quickly. Moreover, WG outputs are set to the observed distributional

properties, primarily on the daily or sub-daily scale [Ailliot et al., 2015]. In this research, we implement the Weather Generator concept on a large spatial-temporal scale and subsequently propose the term Climate Generator (CG).

1.10.2 CG predictors

The climate of a geographic location has various strong dependencies, such as latitude, earth-sun relationships, proximity to large bodies of water, atmospheric and oceanic circulation, topography and local features. The climatological temperature field is relatively smooth, depending most strongly on latitude, surface elevation and continental position. However, the climatological precipitation field is not smooth and has strong longitudinal and non-local dependence. Like a WG, our CG generates a synthetic climatology conditioned on various inputs. In probabilistic terminology, the CG provides a posterior distribution for climate prediction conditioned on the given predictors. Moreover, our CG predicts temperature and precipitation fields jointly, so predictands are correlated with each other. The CG presented herein predicts monthly mean surface temperature and precipitation fields by considering the above characteristic of climate through predictor variables: latitude, longitude, monthly mean sea surface temperature field from a fast low-resolution Energy Balance Model (EBM), surface elevation, ice mask, atmospheric concentrations of carbon dioxide and methane, and orbital forcing. Some predictor variables (latitude, longitude, Carbon dioxide etc.) are already taken into account by the EBM, but perhaps to an inadequate extent. We therefore explicitly include those predictor variables to create our CG and then test whether their inclusion is required via ARD. Latitude and longitude are included in our study because they have direct effects on climate prediction. We select surface elevation as a predictor due to strong vertical temperature gradients that vary laterally. Due to its high albedo, the presence of ice is also included as a predictor. Carbon dioxide and methane data are considered as a predictor given their relatively significant radiative forcing variations over a glacial cycle. Finally, we include monthly mean sea surface EBM temperature in our predictor sets, since the EBM can estimate the sea level monthly mean surface temperature very efficiently. As an attempt to capture non-local effects, the first two EOFs of surface elevation and the area of ice were tested in our CG as predictors but did not yield any significant improvement in CG predictions.

1.10.3 Climate Generator and reality

Computational tools, which are used in climate variability representation, may have many sources of uncertainty. For example, GCMs uncertainties are classified into initial conditions, boundary conditions, parametric and model structure. For the hypothetical case of accurate initial and boundary conditions, and accurate model parameter values, the model structural uncertainty is then isolated: Reality $(t) = GCMs(t) + \alpha$, where α is the structural error. The CG emulates GCMs, so all GCMs uncertainties listed above propagate into the CG. The BANN also estimates its own regressive uncertainty. We make the assumption that this regressive uncertainty is largely due to smaller spatio-temporal scale dynamics and non-local couplings within the GCM and thereby consider it as climate noise. To capture GCM variance, this predictive uncertainty of the BANN is used to specify the variance of uncorrelated Gaussian noise that is added to each CG prediction. This uncorrelated aspect of the injected "climate noise" is a further source of error. We did test the inclusion of the first two surface elevation EOFs in the CG input set, but no significant improvements arose. This CG does not account for the structural error of the GCM (which would be a PhD Thesis in itself).

1.10.4 Present setup and future prospect

The CG is tested for two months: February (the coldest month) and August (the warmest month). The CG is trained with FAMOUS climate model data (rename as CGfamous) for the time period 22 ka to present. To independently test this approach, CG predictions are compared against a 98 kyr interval of GCM (FAMOUS) outputs that was not used for CG training. The CG (BANN) was then retrained with the CCSM (a much better GCM than FAMOUS) on same training interval and rename as CGccsm. The CG is a computationally efficient (simulation of 120 ka to 22.05 ka took 15 minutes, approximately the same computationally speed as the EBM) stochastic climate model. The CG will be coupled with a 3D ice sheet model for glacial cycle modelling. The CG approach could also be applied to more advanced EMICs (e.g. LOVECLIM) for climate generators on short time scales.

1.11 Summary

The objective of this research is therefore:

To create a computationally efficient stochastic climate model (involving in part spatio-temporal downscaling) to simulate atmospheric fields over last glacial cycle timescales that are indistinguishable (for a specified set of metrics) to the output of GCMs.

1.12 Thesis Overview

This thesis is written in MUN manuscript format. This chapter provides a review of the literature and background materials for the article that is presented in chapter two. More CG (BANN) outouts comparison with FAMOUS, CCSM and EBM climate model outputs are added in the Appendix.

Chapter 2

Stochastic climate representation for millennial scale integration over North America

2.1 Abstract

This paper presents a computationally efficient stochastic climate model to simulate atmospheric fields (specifically monthly mean temperature and precipitation) on large spatial-temporal scales. In analogy with Weather Generators (WG), the model can be considered a "Climate Generator" (CG). The CG can also be understood as a field-specific General Circulation climate Model (GCM) emulator. It as invokes aspects of spatio-temporal downscaling, in this case of a T21 Energy Balance climate Model (EBM) to a GCM scale. The CG produces a synthetic climatology conditioned on various inputs. These inputs include sea level temperature from a fast low-resolution EBM, surface elevation, ice mask, atmospheric concentrations of carbon dioxide, methane, orbital forcing, latitude and longitude. Bayesian Artificial Neural Networks (BANN) are used for nonlinear regression against GCM output over North America. Herein we detail and validate the methodology. To impose natural variability in the CG (to make the CG indistinguishable from a GCM) stochastic noise is added to each prediction. This noise is generated from a normal distribution with standard deviation

computed from the 10% and 90% quantiles of the predictive distribution values from the BANNs for each time step. This derives from a key working assumption/approximation that the uncertainty in our prediction is in good part due to the "noise" of the GCM climate. Our CG is trained against GCM (FAMOUS and CCSM) output for the last deglacial interval (22 ka to present year). For predictive testing, we compare the CG predictions against GCM (FAMOUS) output for the disjoint remainder last glacial interval (120 ka to 22.05 ka). The first application of the CG will be for glacial cycle modelling with the Glacial Systems Model.

2.2 keywords

Climate Generator, Weather Generator, Climatology, BANNs

2.3 Introduction

A Weather Generator (WG) generates a synthetic time series of weather data for a location based on the statistical characteristics of the observed weather. A WG generally operates on a small scale, is computationally efficient to enable numerous random realizations, and its output is designed to have the same distributional properties as the observed time series, usually on the daily or sub-daily scale [Ailliot et al., 2015]. In contrast, climate models reproduce the behaviour of the whole atmosphere and its interactions with the other components of the Earth system (oceans, vegetation, etc.) on a large spatial scale and generally for longer-term intervals. Here, we expand the WG concept to a large spatial-temporal scale, proposing the term Climate Generator (CG) for the expansion.
Our primary aim is to create a fast, efficient stochastic climate representation for glacial cycle scale modelling. General Circulation Models (GCMs) are presently too computationally expensive for such contexts. For example, the widely used Community Climate System Model (CCSM3) took about 2 years on 100 cores to simulate the 22 ka to present [Liu et al., 2009]. Earth System Models of Intermediate Complexity (EMICSs) are more appropriate for ten kyr scale climate simulations [Claussen et al., 2002], but with varying tradeoffs between resolution, accuracy of climate representation, and computational speed. Energy Balance climate Models (EBMs) are the fastest EMICS for a given resolution but, due to their lack of atmospheric dynamics and precipitation fields, are inadequate for glacial cycle modelling [Hyde et al., 1990], [Tarasov and Peltier, 1997]. To date, most glacial cycle

ice sheet simulations use a glacial index (for example Tarasov et al. [2012]) in a combination with LGM timeslice GCM output for their climate forcing.

Unless full energy-balanced surface mass-balance is being computed, ice-sheet models for glacial cycle contexts do not require accurate year to year prediction in their climate forcing. The dynamical memory of ice sheets filters interannual fluctuations. As such, 30 to 50 year scale monthly mean climatologies with statistics for shorter scale temperature variability are a reasonable trade-off for enabling computation over such timescales. Given this context, we invoke the concept of a Climate Generator (CG) for improving the output of fast EMICS (in this case a geographically resolving 2D EBM) for 100 kyr scale contexts. The CG is a large spatio-temporal scale dynamic representation of climate based on a regressed relationship between the EBM (and various other predictor variables) and relevant GCM outputs.

To build up such a relationship, Artificial Neural Networks (ANNs) are common in small spatial-temporal scale climate prediction [Schoof and Pryor, 2001]. ANNs have the potential for complex non-linear input-output mapping [Dibike and Coulibaly, 2006]. However, ANNs do not have associated uncertainty estimates, and over-fitting is a hazard. To minimize over-fitting and to find an optimum network, ANNs rely on a cross-validation test. Cross-validation does not use training data efficiently as it requires disjoint data sets for testing and parameter estimation. Bayesian Artificial Neural Networks (BANNs) generate uncertainty estimates and avoid the need for cross-validation. In BANNs, an assumed prior distribution of parameters (weight and biases) is used to specify the probabilistic relationship between inputs and outputs. The prior distribution is updated to a posterior distribution by a likelihood function through Bayes theorem. The predictive distribution of the network output is acquired by integration over the posterior distribution of weights. BANNs are used in different applications e.g., to create weather generators [Hauser and Demirov, 2013], for model calibration [Hauser et al., 2012], and for short time scale climate prediction [Maiti et al., 2013], [Luo et al., 2013]. Our CG uses BANNs to estimate a posterior distribution for climate prediction/retrodiction conditioned on various inputs including the output of an EBM.

2.4 Test study region and data

The test study region of interest is North America (including Greenland; more specifically, longitude 188E : 355E and latitude 34N : 86N). This combines a continent that experience past ice cover but no significant present-day ice cover with a region that has had continued ice cover to present. We

train our CG (CGccsm) against the full-time interval of the CCSM3 (Monthly means, T31) climate model [Liu et al., 2009] radiative surface temperature and precipitation monthly mean output. For methodological validation, we use the FAMOUS (low-resolution GCM, atmospheric part of the model has resolution $5^\circ \times 7.5^\circ$) climate model data sets [Smith and Gregory, 2012] (1.5 meter air temperature and precipitation) and train our CG (CGfamous) against the same time interval as CCSM. The FAMOUS model was run with an accelerated mode (factor 10) for the full Last Glacial Cycle (LGC) (120 ka to present). We divide the full-time interval into two parts: the training interval is 22 ka to the present year and the test interval is 120 ka to 22.05 ka. For this initial test of concept, the CG is implemented for the coldest month (February) and the warmest month (August), with 50 year climatologies (5 consecutive years for FAMOUS and 5 years spaced 10 years apart for CCSM) for model calibration and validation. Each month has 2400 time steps for FAMOUS (120 ka to present year) and 440 time steps for CCSM3 (22 ka to present year). Each time step has 231 (21x11) gridcells (FAMOUS) and 602 (43x14) gridcell (CCSM) data (for each predictors) and our CG is trained to predict on each gridcell. For model validation, we compare the CGfamous output directly to that of FAMOUS over the test interval. These data sets are not used for the CGfamous training. For further evaluation, we compare our CGfamous, CGccsm outputs with FAMOUS, EBM (T11 @1500 km) and CCSM3 predictions over the last glacial (test) period. For the comparisons, EBM output [Deblonde et al., 1992] sea level temperature is adjusted to surface temperature with a lapse rate of 6.5 K/km, as this has been standard practice for using EBM climate models. EBMs generally do not take into account the spatio-temporal variation of vertical temperature gradients.

2.5 Methods

The CG uses Bayesian Artificial Neural Networks (BANNs) for estimating an evolving climate state as a function of various inputs. The BANNs also estimate predictive uncertainty which (in an arguable leap) we take to represent the shorter scale un-resolved variability in the climate. BANNs are effectively a set of artificial neural networks with individual parameters from a posterior probability distribution derived from training the network against observed input-output sets. The CG estimates target values based on the mean from the resulting set of networks, and its squared error. Further parameter sampling results from creating several network sets with distinct initial seeds.

2.5.1 BANN design and training

We design our ANN architecture by using the software for flexible Bayesian modelling package (freely available) at http://www.cs.utoronto.ca/radford/fbm.software.html. Different architectures (number of hidden layers, node size, connection of inputs, hidden units and outputs) and different predictors are first tested. Architectures are selected according to the predictive skill on the test data. Step size and prior specification are then adjusted to improve prediction capability. To choose an appropriate predictor set, we tested the following predictor variables: latitude, longitude, EBM sea level temperature (Deblonde et al. [1992]), carbon dioxide (Luthi et al. [2008]), methane (Loulergue et al. [2008]), surface elevation [Smith and Gregory, 2012], surface type (ice) [Smith and Gregory, 2012], orbital forcing (came from EBM) in different time and location, temperature and precipitation ([Smith and Gregory, 2012] and [Liu et al., 2009]), melt water flux (five different locations; [Smith and Gregory, 2012]), first two EOFs of ice surface elevation, and ice area. The outputs are monthly mean surface temperature and precipitation. Various network architectures and different combinations of predictor variables were tested. Automatic Relevance Determination [ARD] Neal [2012] was used to identify which predictors provide meaningful weight in the distribution value. Various combinations of the input set and network architectures were also evaluated against the test interval subset of the GCM output (as detailed below). The resultant optimal input set is comprised of: latitude, longitude, surface elevation, ice, carbon dioxide, methane, orbital forcing (June/July/August mean solar insolation at 60N), and EBM sea level temperature. The best fitting BANN architecture has two hidden layers of $tanh(x)$ hidden unit as detailed in Figure 2.1.

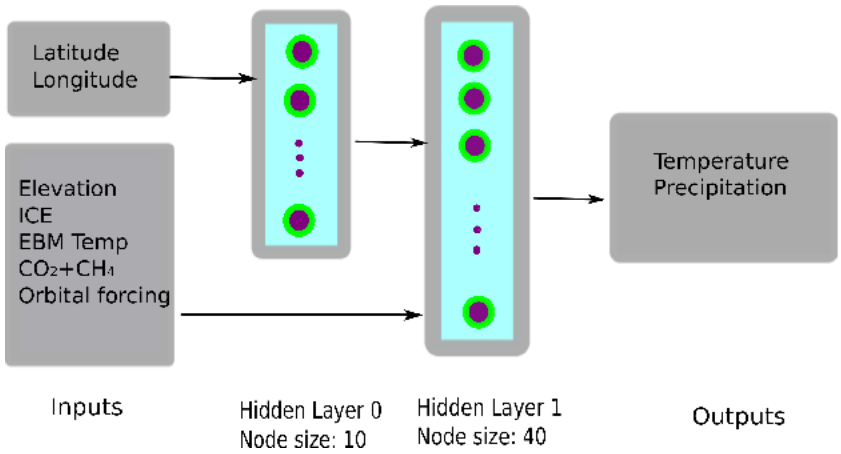


Figure 2.1: Final ANN Diagram (Architecture type I) used in CGfamous and CGccsm prediction

2.5.2 BANN Implementation

Bayesian Artificial Neural Networks (BANNs) estimate a probability distribution and are derived by training the network against available data using Bayesian inference. Markov Chain Monte Carlo (MCMC) sampling is used for selecting the distribution parameters for the networks. The step by step procedure for BANN's implementation follows Neal [2012]:

- Process predictor and predictand data sets.
- Define architecture and specify model to have real valued targets.
- Define a prior distribution for parameters (weight and biases). The prior is specified hierarchically for a group of parameters with a three-level approach as follows. Each prior is sampled from a Gaussian distribution with zero mean and some precision (inverse standard deviation). The precision for a "sub-group" of parameteris in turn selected from a Gamma distribution with a selected shape parameter and with a mean given by a hyperparameter. Finally, this hyperparameter is sampled from a Gamma distribution with a specified mean and with another assigned shape parameter. Mean precision at the top level is assigned as 0.1 (width) and the shape parameters of the Gamma distribution are assigned a value of 2. Priors for input to hidden layer, connections between hidden layers and hidden layers to outputs are automatically re-scaled based on the number of hidden units and the prior of the output bias is specified as a Gaussian prior with mean zero and standard deviation 10.

- A noise model is fitted to estimate network parameters. To fit a noise model each prediction for targets is considered to be a sample from a Gaussian distribution. The noise levels specification follows the same three level approach used for specification of the priors. The targets are modelled as the network outputs plus Gaussian noise.

- Specify data for training and testing.

- Initialize the network with hyperparameters set to (say) 0.3 and all parameters set to zero. Markov chain operations are defined, where each iteration consists of order fifty repetitions of the following steps: Gibbs sampling for the noise level, hybrid Monte Carlo sampling using a fixed trajectory and step size adjustment factor. In this stage, the hyperparameters are not updated and remain fixed at a value 0.3.

- A single iteration of the above process is representative of one step in Markov chain simulation. The rejection rate is examined after a number of (say 50) hybrid Monte Carlo updates. If the rejection rate is high (say, over 40%), the Markov chain simulation is repeated with a smaller step size adjustment factor.

- a network is stored in the log file containing the parameters and hyperparameters values. Markov chain sampling is repeated and overrides the previous set. Each iteration consists of say five repetitions of the following steps: Gibbs sampling for both the hyperparameters and the noise level followed by hybrid Monte Carlo sampling as above. (A long trajectory length is useful for the sampling phase).

- By looking at the hyperparameters values and quantities such as the squared error on the training set, we can get an idea of when the simulation has reached equilibrium or not. After that, we can start prediction.

- Generate predictions (mean, 10% and 90% quantiles) for the test cases from the resultant distribution of networks. By using different initial seeds, ensembles of several networks are generated by sub-sampling from the later segments of the Markov Chains.

The detailed implementation procedure is given in Neal [2012].

2.5.3 Adding noise

Gaussian noise is added in our CG prediction to account for (at least in part) seasonal to decadal climatic variability not captured by the EBM. This natural variability (noise) is physically correlated across space and time. But given the context (coupling with ice sheet models for glacial cycle timescales) and for computational simplicity, the CG noise injection uses uncorrelated random sampling. Ice sheet thermodynamic response to climate is smoothed to centennial or longer timescales. Surface mass-balance response for the given grid scales will be sensitive to the variance of temperature but not to spatial correlations nor much to temporal correlations. Correlations between temperature and precipitation could have significant impact, especially during the potential melt season. The August residual (CG without noise - FAMOUS) correlation map between the temperature and precipitation fields shows magnitudes of mostly less 0.3 (Fig. A.18 in the Appendix) and are therefore relatively small (this does not, however, rule out significant non-linear relationships). The random noise is added to each time step of our CG predictions by generating a random sample from Gaussian distribution with $\sim N(\mu = 0, \sigma)$, where μ = Mean and σ = Standard deviation (80% confidence interval scale). The standard deviation is computed through the BANNs predicted 10th percentile and 90th percentile of the predictive distribution of a single guess for each case. Standard deviation is computed from the following Equation:

$$\sigma = \frac{(X_{90\%} - (X_{90\%} + X_{10\%})/2)}{Z_{90\%}} \tag{2.1}$$

where $Z_{90\%} = 1.28$ (Z values or score), calculated from the statistical table. The values of σ, defined in the Equation 2.1 had space and time dependence. This assumption that BANN predictive uncertainty can provide an approximate estimate for the unresolved climatic variability is tested in part below.

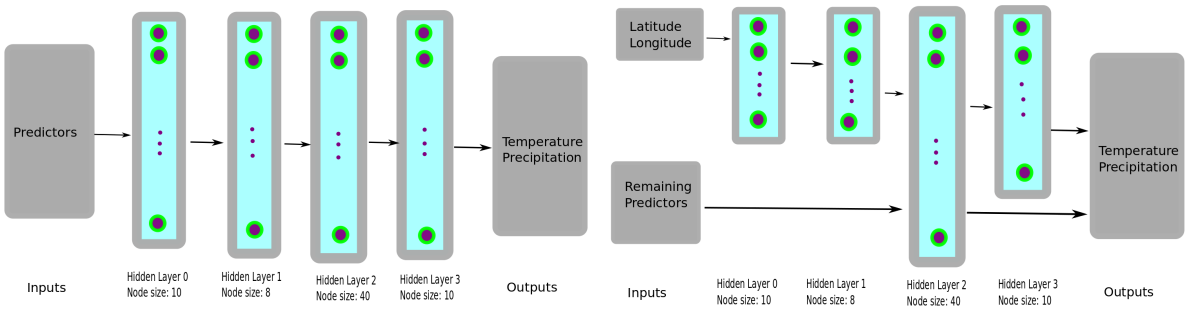
2.6 Implementation Results

This research implements the "Turing Test" concept as a "Climate Turing Test (CTT)" to measure the prediction capability of our CG. The CTT determines whether or not CGfamous is capable of predictions like the FAMOUS climate model. To implement the CTT concept, a direct comparison was done between CGfamous and FAMOUS Climate model outputs over the test interval (120 ka

to 22.05 ka) using characteristics such as Root Mean Square Error (RMSE), visual patterns in map plots and Empirical Orthogonal Function (EOF) projection to measure the difference in outputs between CGfamous and FAMOUS relative to that between CCSM and FAMOUS. Here we give some example comparisons for different architectures and predictor sets.

2.6.1 Selection of BANN architecture

More than 100 different BANN architectures (different number of hidden layers, different connections and node sizes) with different combinations of predictor sets were tested. To convey the sensitivity to architecture and predictor set, we present results for three basic architectures (Figure 2.1 and Figure 2.2) in combination with various predictor sets (Table 2.1). Architecture type I (Figure 2.1) gave the best overall fit of CGfamous to FAMOUS over the test interval.



(a) Architecture type II (b) Architecture type III

Figure 2.2: ANN Diagrams are tested to improve CG prediction

Table 2.1: Predictor sets for displayed CG results

Combination 1: PS1 = Latitude, Longitude, surface elevation, ice, EBM temp
Combination 2: PS2 = PS1, CO_2 + CH_4, orbital forcing, melt water flux
Combination 3: PS3 = PS1, carbon dioxide, methane, orbital forcing
Combination 4: PS4 = PS3, EOF-1 and EOF-2 of ice data, ice volume

Table 2.2: Different models based on architecture and inputs

Model Name	Predictors	Architecture type
A	PS3	II
B	PS1	I
C	PS3	III
D	PS2	I
E	PS4	I
F	PS3	I

Six model names are assigned in Table 2.2 based on different predictor sets and architecture (Table 2.1).

Table 2.3: MEAN DIFFERENCE (MD) and RMSE relative to the FAMOUS (except for last entry) over the full grid area (train period)

	TEMP (DEG C)				PREC (cm/month)			
	FEB		AUG		FEB		AUG	
Model	MD	RMSE	MD	RMSE	MD	RMSE	MD	RMSE
A	3.91	9.07	4.09	5.7	0	3	1	3
B	2.85	10.19	-2.93	9.9	1	4	2	5
C	2.43	7.94	5.45	5.36	1	3	1	4
D	-3.3	4.13	-1.73	2.67	0	2	0	2
E	-2.35	5.77	-1.05	4.6	0	2	1	3
F	-0.45	3.92	-1.23	2.52	0.5	3	0	3.1
M (1)	5.87	10.51	-4.4	4.29	0.02	3.08	0	3.06
CCSM (2)	5.86	10.99	-4.4	4.08	0.5	4.1	-1.3	4.1
EBM	12.46	11.0	-1.52	4.77				
(1) versus (2)	-0.02	4.14	0.01	2.67	0.5	4.1	-1.3	4.1

These models are compared with FAMOUS model outputs over the training interval in Table 2.3 and over the test interval in Table 2.4. The positions of each letter appearing in Figure 2.3 and 2.4 quantifies how closely that model's simulated temperature and precipitation pattern match the FAMOUS climate model outputs and gives a graphical summary of comparisons of the RMSE and Standard deviation. RMSE is computed from the differences between different CG predictions with FAMOUS climate model outputs. The Standard deviation of FAMOUS is indicated by the black contour line in Taylor diagrams. Our main selection criterion is the minimization of RMSE. Beside this, mean deviation (general bias) comparison on Table (2.4) and standard deviation and correlation from Figure 2.3 and Figure 2.4 allow us to see how comparable our models prediction are with FAMOUS. The central RMSE is about 2.98 °C (August temperature) and 5.49 °C (February temperature) for model F, which are the lowest compared to all other models in Table 2.4. In the case of precipitation, the RMSE of model F is about 1.9 cm/month (February) and 2.8 cm/month (August), which also are the least compared to all other models.

Table 2.4: MEAN DIFFERENCE (MD) and RMSE relative to the FAMOUS (Test period) over the full grid area

	TEMP (DEG C)				PREC (cm/month)			
	FEB		AUG		FEB		AUG	
Model	MD	RMSE	MD	RMSE	MD	RMSE	MD	RMSE
A	4.45	8.08	3.93	9.1	-0.16	2.5	1.07	4
B	3.35	9.11	-3.09	8.5	0.83	3	2.07	5
C	3.2	6.49	5.29	12.5	-0.18	5	0.07	3
D	-2.75	5.74	-1.29	5.22	-0.16	2.3	-0.93	3
E	-1.75	5.38	-1.2	4.68	-0.15	2.3	0.07	3.4
F	-2.45	5.49	-1.45	2.98	-0.15	1.9	0.07	2.8
M	-4.26	11.27	-6.64	5.36	0	3	0.03	3.3
EBM	10.82	10.45	-5.3	4.76				

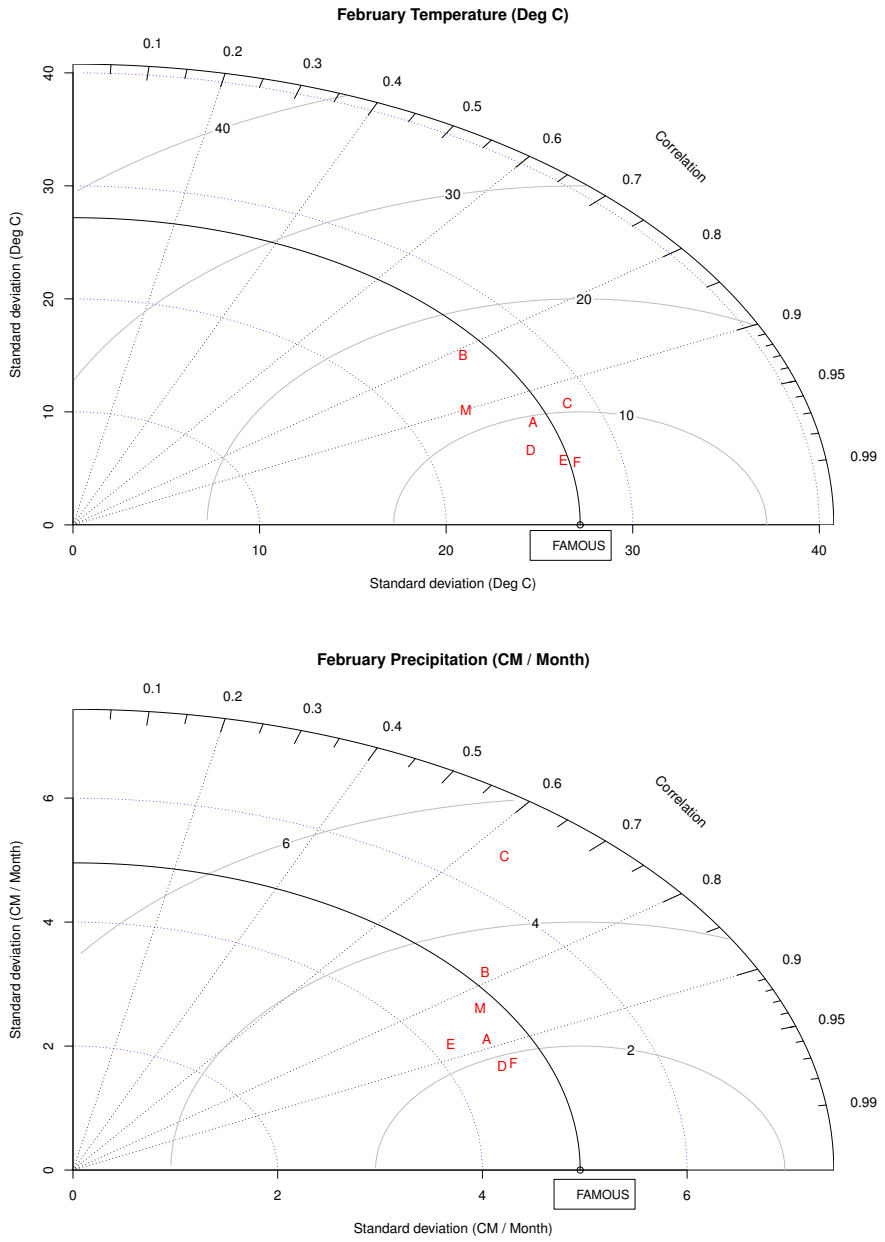


Figure 2.3: Taylor diagram displaying a statistical comparison with FAMOUS (test part) of six different CG estimates. Pattern similarities are quantified based on their correlation and centered root mean square difference between CG and FAMOUS, and standard deviation with respect to the mean of the corresponding field. Contour grey lines indicate the root mean square (RMS) values. The model M is the same as model F but trained with CCSM climate model data.

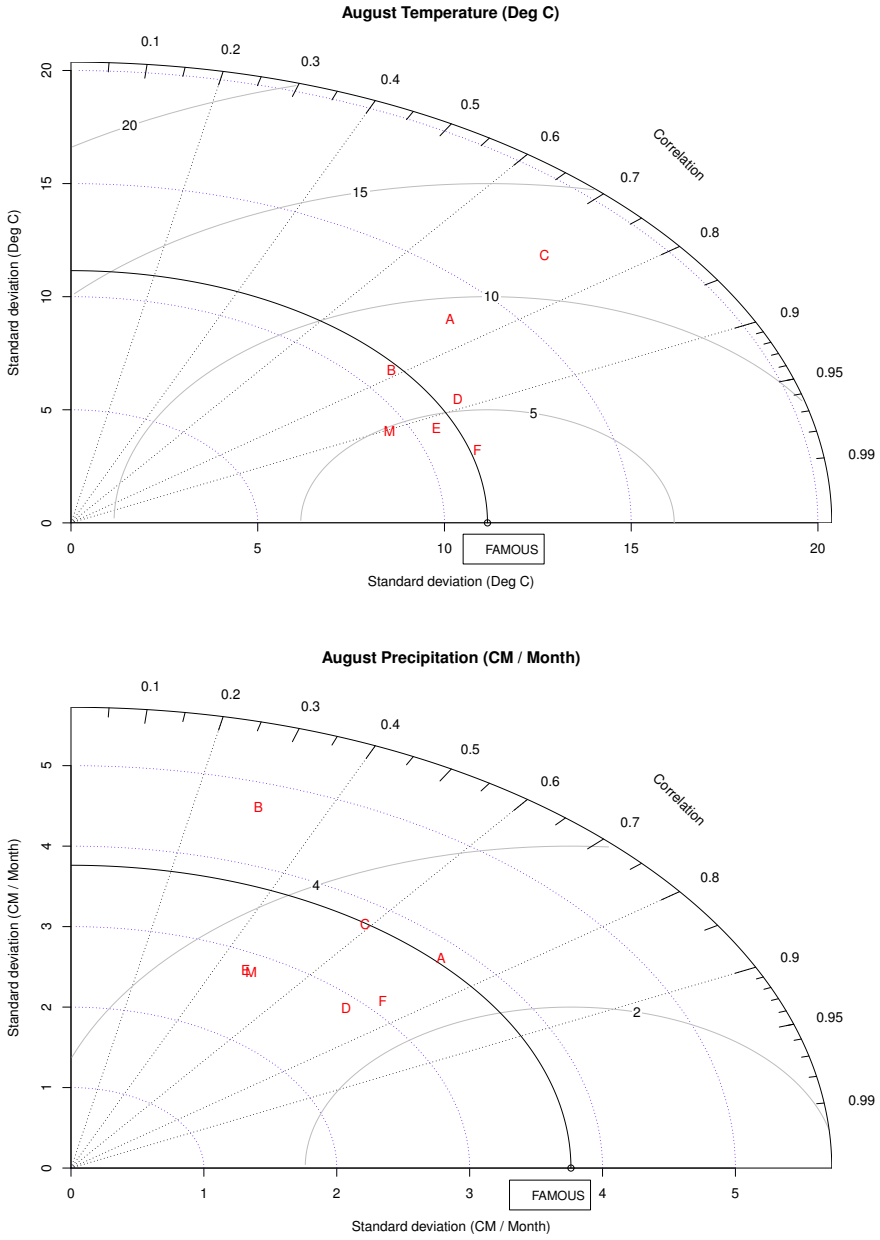


Figure 2.4: Taylor diagram displaying a statistical comparison with FAMOUS (test part) of six different CG estimates. Pattern similarities are quantified based on their correlation and centered root mean square difference between CG and FAMOUS, and standard deviation with respect to the mean of the corresponding field. Contours grey line indicates the root mean square (RMS) values. The model M is the same as model F but trained with CCSM climate model data.

The model F has strong mean correlation (over space and time) with FAMOUS at about 0.98 for February temperature and 0.92 for February precipitation in Figure (2.3 and 2.4). The model F has the best fit compared to other models listed in Table 2.4. As is evident in Tables 2.3 and 2.4 and figures (2.3 and 2.4), there is significant sensitivity to network architecture and predictor set with a factor 4 range in temperature RMSE and factor 2 range in precipitation RMSE for the example combinations. The sensitivity to predictor sets is quantified in more detail through ARD as described below.To choose predictor sets, an ARD analysis has been done. The ARD test helped us to choose which inputs are relevant for the outputs and is determined based on the hyperparameters values which control the standard deviation for weights and biases in different groups. In Neal [2012], these hyperparameters values are referred to as "Sigma" values. The significance of each input is non-linear proportional to the Sigma value in Table 2.5.

Table 2.5: ARD analysis for the Model F (CGfamous)

Predictors	Sigma value
Latitude	1.30
Longitude	0.67
Elevation	2.85
Ice	2.79
CO2+CH4	1.18
Orbital forcing	3.42
EBM temp	4.46

For the case of Model F, EBM temperature is the most significant input. However, even though the EBM computes orbital forcing and accounts for greenhouse gases, orbital forcing is still the next most significant input. Longitude is the least important, even though temperature and precipitation have in reality high dependence on longitude given atmospheric circulation dependencies. The EBM does have a slab ocean with thermodynamic sea ice, and this result suggests that continentality effects might be reasonably captured by the Model EBM. Further comparisons were carried out with field plots and EOF projections for all simulated fields. Model F predictions have the least

RMSE (compared to FAMOUS) among all tested models. It has the highest positive correlation and comparable mean difference (with FAMOUS). Therefore the model F has been chosen and renamed CGfamous.

2.6.2 Model Comparison

Weather Generator performance can be tested against new observations, but such opportunities are limited in our case. The CG is evaluated based on computing statistics (RMSE, Mean deviation, correlation), the goodness of fit (including noise levels) and qualitative consideration against FAMOUS climate model outputs (test interval). The best and worse fits are identified between CG prediction and FAMOUS outputs on specific regions or latitude bands, winter versus summer and full grid area versus ice region. The comparison over the ice region gives the opportunity to check the prediction capability of our CG in the context of glacial cycle modelling. Criteria such as space-time scale appropriateness, patterns, and climate noise variability at shorter time scales are introduced to measure variance. The difference between CCSM and FAMOUS over the training interval is taken as a minimum value of model uncertainty and thereby our reference misfit bound for the climate Turing Test.

Over both the test and training interval, the RMSE of CGfamous simulated temperature field (about FAMOUS) is about 50% less than the RMSE of CCSM and EBM model (about FAMOUS) over the full grid area. For precipitation over the test interval, the RMSE of CGfamous (about FAMOUS) is approximately 42% less than the RMSE of CCSM (CCSM versus FAMOUS). The CG simulated August temperature has a better fit with FAMOUS (less RMSE) compared to that of February. Also, CG seems to work better in the ice region (approximately 54% or more less RMSE compare to the full grid area). Besides these quantitative statistics, CGfamous also has improved the geographic pattern of misfits (CGfamous -FAMOUS versus FAMOUS - CCSM and FAMOUS - EBM). The CGfamous simulated temperature field (both months) has a clear cold bias (in test part) compared to FAMOUS. But CG CCSM has a sharp and warm bias (February) in addition to a clear cold bias (August) compared to the FAMOUS test map. The CG simulated temperature field has captured approximately 70% temporal variance (based on the leading two EOFs) of that of FAMOUS. Precipitatation is a challenge for all models, and the CGfamous precipitation field captures about only 40% (based on leading two EOFs) of the temporal variance compared to FAMOUS. For context, the CCSM precipitation field is also far from that of FAMOUS.

For both months over the training period, the CGfamous (network F) temperature RMSE relative to FAMOUS is less than half of our structural uncertainty reference (i.e. CCSM - FAMOUS) and the corresponding precipitation RMSE is about 33% smaller (TABLE 2.4). The temperature RSME not unexpectedly increases over the predictive test region, but critically these values are still about half of the CCSM - FAMOUS reference RMSE. CGfamous temperature is highly correlated with FAMOUS (0.94 or higher) over the test interval with a temperature bias (MD) that is 2 degree (about 40%) lower than the CCSM bias over the test interval (Table 2.4). As structural uncertainty is larger than the difference between CCSM and FAMOUS, CGfamous temperature largely passes the Climate Turing Test for these metrics. The approximate factor 3 improvement in RMSE of CGfamous versus that of the EBM validates the CG role of statistical improvement of the fast simplified models.

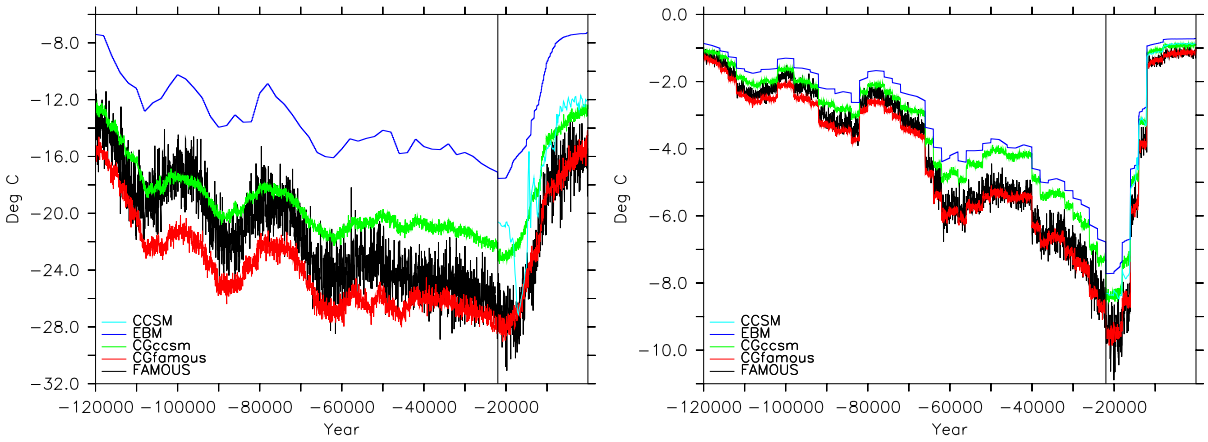
Table 2.6: Mean Difference (MD), Correlation (COR) and RMSE relative to FAMOUS (except for last entry) over the ice region (Train period)

	TEMP (Deg C)						PREC (cm/month)					
	FEB			AUG			FEB			AUG		
Model	MD	RMSE	COR	MD	RMSE	COR	MD	RMSE	COR	MD	RMSE	COR
EBM	1.58	4.41	.99	-0.38	1.76	.98						
CGfamous	0.11	1.33	1	0.02	0.53	1	0.05	0.41	.87	0.01	0.92	.95
CGccsm(1)	1.06	3.40	.99	-1.23	1.24	.99	0.06	0.38	.86	0.01	0.89	.96
CCSM(2)	1.06	3.58	.99	-0.27	1.19	.99	-0.01	0.39	.87	-0.16	0.78	.96
(1) versus (2)	0	0.92	.99	-0.02	0.51	1	0.02	0.30	.86	0.16	0.54	.96

Table 2.7: Mean Difference (MD) , Correlation (COR) and RMSE relative to FAMOUS (Test period) over the ice region

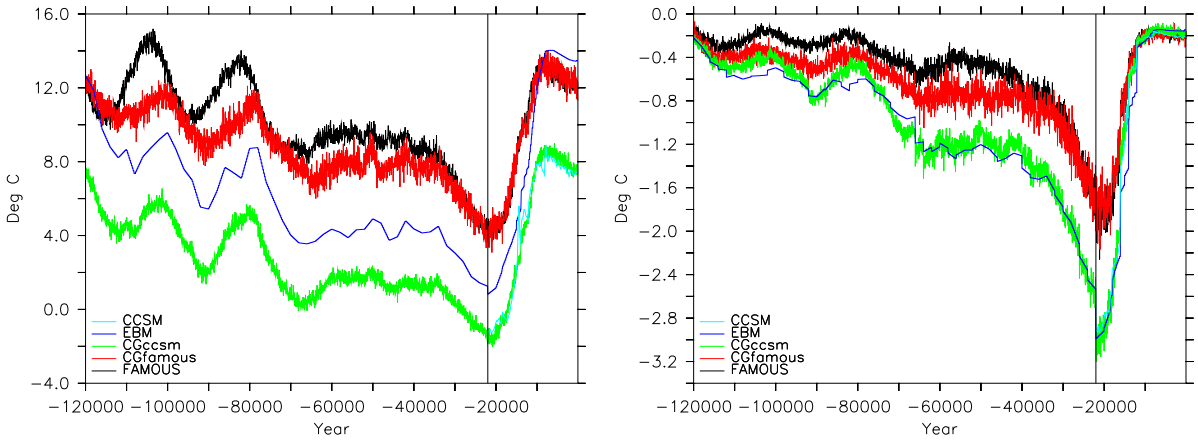
	TEMP (Deg C)						PREC (cm/month)					
	FEB			AUG			FEB			AUG		
Model	MD	RMSE	COR	MD	RMSE	COR	MD	RMSE	COR	MD	RMSE	COR
EBM	1.68	4.24	.99	-0.83	2.21	.93						
CGfamous	-0.37	1.79	.99	-0.26	1.10	.96	0.02	0.40	.75	-0.20	0.94	.89
CGccsm	1.02	3.2	.98	-0.68	1.94	.94	0.02	0.31	.77	-0.08	0.78	.88

For precipitation over the test interval, the RMSE values of CGfamous are about 53% smaller (February: 1.9 versus 0.41 cm/month) and 31% smaller (August: 2.8 versus 4.1 cm/month) than the RMSE of CCSM precipitation (Table 2.4). Except for the weaker correlation for August compared to February (0.72 versus 0.92 cm/month) over the test interval (arguably again within structural uncertainty), CGfamous precipitation also passes this components of the climate Turing test.



(a) February (Full grid area)

(b) February (Ice region)



(c) August (Full grid area)

(d) August (Ice region)

Figure 2.5: Comparison of the spatial mean (with latitudinal weighting) temperature time series. The black vertical line separates the test (left) and training part (right).

Even with the increased complexity and higher resolution of CCSM, there is little deterioration in the training fit of CGccsm to CCSM compared to that of CGfamous to FAMOUS. All statistics (RMSE, MD and correlation) for CGfamous and CGccsm are better when computed just over ice covered regions (Table 2.6 and Table 2.7).

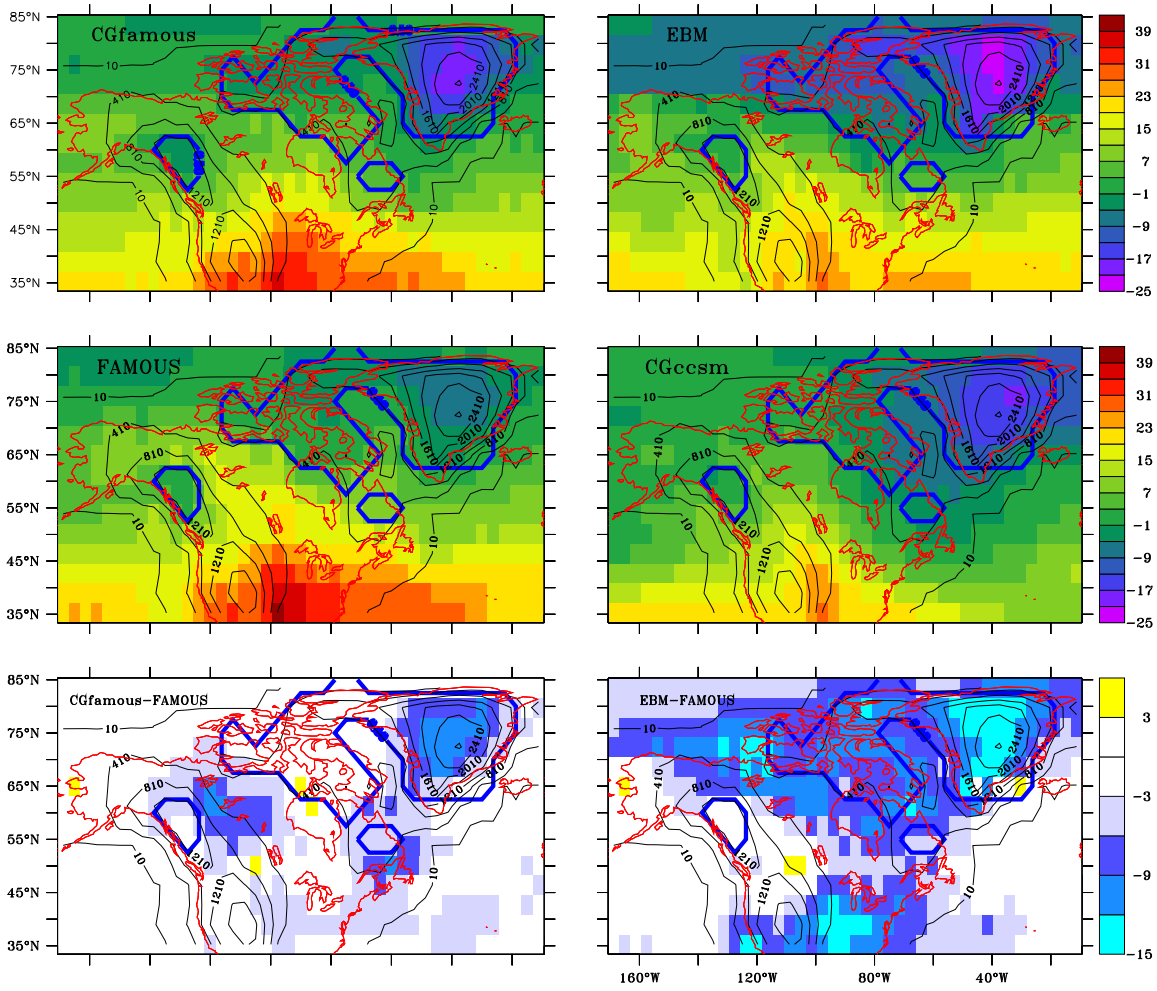


Figure 2.6: The August temperature field (Deg C) at 100 ka (1st and 2nd row) with elevation and ice contours shown in black and blue. The difference between plots are shown in the 3rd row. Model names are indicated in the top left corner in each box.

Our climate Turing test also requires assessment of map plot timeslices. A 100 ka August temperature fields comparison (Fig. 2.6) indicates regional biases, with the most evident being a strong

cold bias over Greenland. However, the discrepancies are significantly less than the 18 ka difference between CCSM and FAMOUS (Fig. A.4 in the appendix). Furthermore, there is no obvious visual pattern that one could apriori use to ascertain which field was from FAMOUS versus CGfamous.

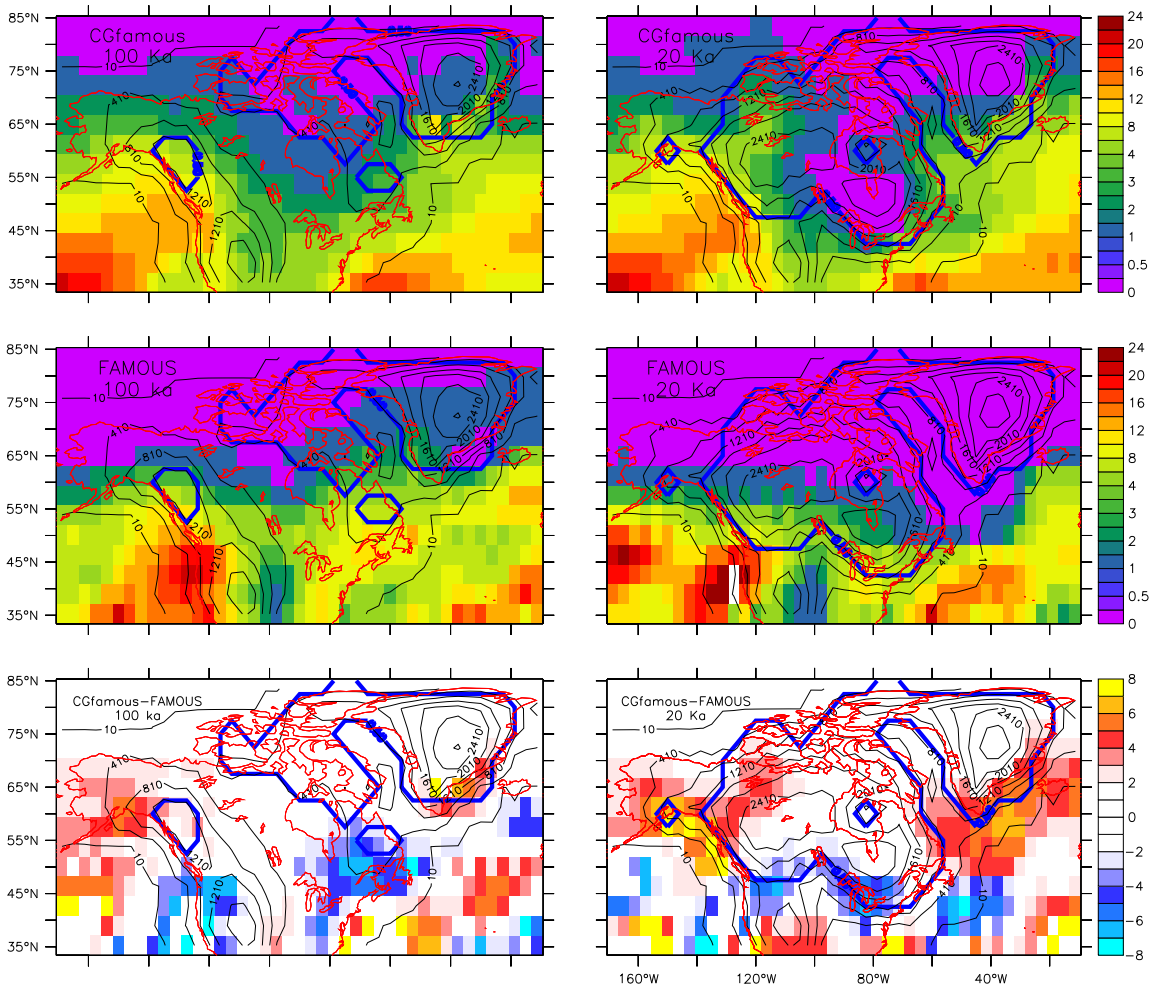


Figure 2.7: February precipitation field (cm/month). Left column: test case. Right column: Training Case. Model names and times are indicated in the top left corner in each box.

The geographic pattern of precipitation misfit between CGfamous and FAMOUS at 18 ka is very close to that of CCSM and FAMOUS (Fig. A.5). The misfit patterns for example instances from predictive and training regimes (100 ka and 20 ka in Fig. 2.7) do not show obvious poorer predictive

capability of the CG for the 100 ka timeslice than for the 20 ka training timeslice.

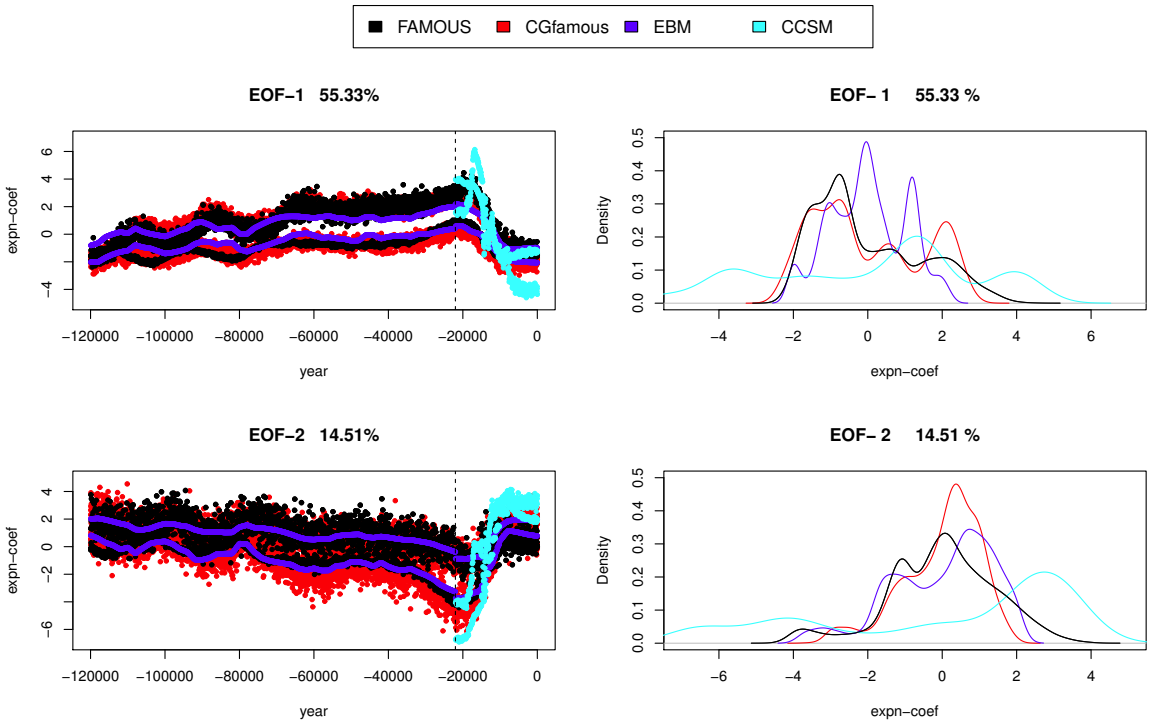


Figure 2.8: Temperature field, left column: Distribution of the expansion coefficient over total time steps for the leading two EOFs of the FAMOUS Model data (black), and the distribution of the expansion coefficients over time obtained by an ensemble projection of CGfamous simulation (red), EBM temperature (blue) and CCSM (cyan) onto the same EOFs. Top time series represent the August EOFs and bottom time series are the February EOFs. Right column: Distribution of the expansion coefficients for the leading two FAMOUS EOFs for the FAMOUS (black), CGfamous (red), EBM temperature (blue), and CCSM (cyan) datasets

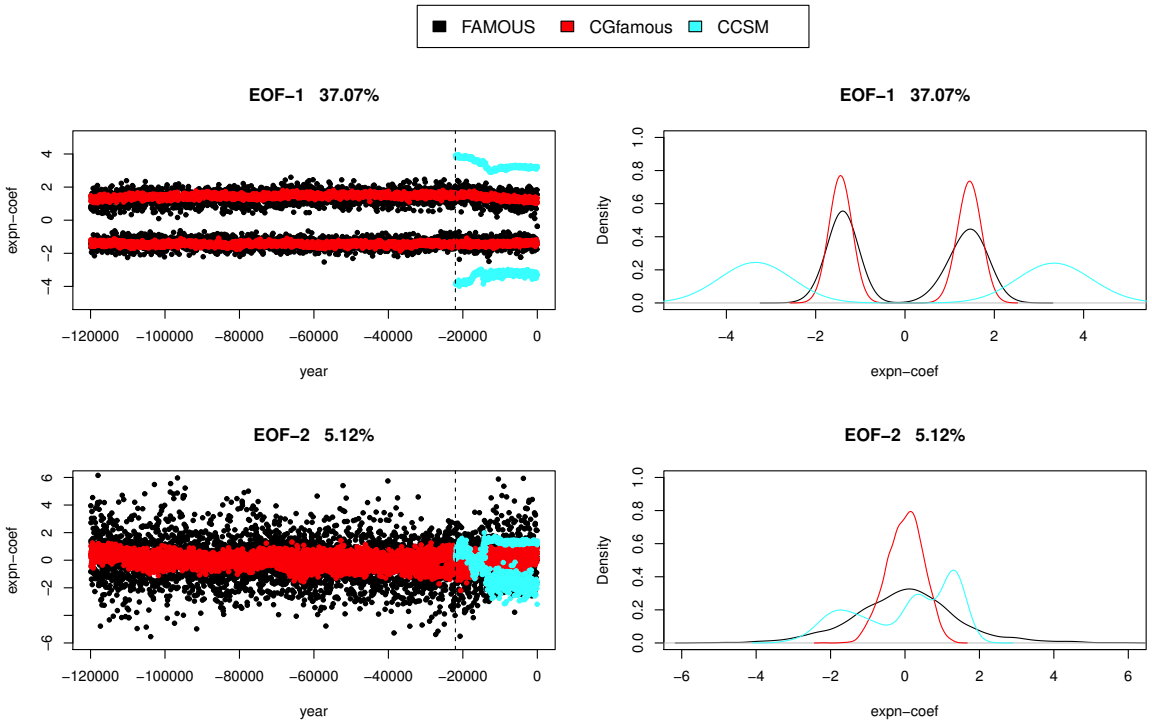


Figure 2.9: Precipitation field, left column: Distribution of the expansion coefficient over total time steps for the leading two EOFs of the FAMOUS Model data (black), and the distribution of the expansion coefficients over time obtained by an ensemble projection of CGfamous simulation (red) and CCSM (cyan) onto the same EOFs. Top time series represent the August EOFs and bottom time series are the February EOFs. Right column: Distribution of the expansion coefficients for the leading two FAMOUS EOFs for the FAMOUS (black), CGfamous (red) and CCSM (cyan) datasets

For spatial comparison over time, the CCSM simulated fields and an ensemble of CGfamous simulations are projected onto the two leading EOFs of the FAMOUS climate model (Figure 2.8 for temperature and Figure 2.9 for precipitation). CCSM simulated fields have less variance compared to FAMOUS and the CGfamous simulated temperature field has better fit with FAMOUS compared to CCSM. The time evolution of the expansion coefficients in the CGfamous simulated fields compared

to FAMOUS are comparatively better than that of CCSM. The 1st EOF represents about 56% of the simulated CGfamous temperature field, while the 2nd EOF represents about 15% (Figure 2.8). For the case of precipitation, only the first EOF is significant (37%), again with a closer match to FAMOUS than that of CCSM.

In summary, the CGfamous simulated temperature field better fits FAMOUS compared to CCSM versus FAMOUS and EBM versus FAMOUS and efficiently generates ensembles which represent large-scale climate variability. Even though the EBM has no precipitaition field, CGfamous is still able to capture FAMOUS precipitation fited to our structural error (CCSM-FAMOUS) reference. The above comparisons have led our CG to reasonably pass the Climate Turing Test.

2.7 Discussion

We have constructed a computationally efficient climate model (our Climate Generator or GCMs emulator) as an alternative to expensive GCMs for 100 kyr scale integrations. To compensate for inadequate variance in the BANN output within the CG, Gaussian noise is injected into our CG at each time step with zero mean and spatially varying standard deviation calculated from ensemble networks prediction values (10% and 90% percentiles). The CG takes about 15 minutes to generate February and August climatologies over the 120 ka to 22 ka interval at 50 year time steps. To compare the prediction capability of our CG with the GCMs, we introduced the Turing test concept as a Climate Turing Test (CTT). To implement the CTT concept, a direct comparison was done between CGfamous and FAMOUS Climate model outputs (120 ka to 22 ka). These data sets are not used in CG training. The difference between the CCSM and the FAMOUS over the training interval (20 ka to present year) are considered as a minimum structural uncertainty estimate for both GCMs. We take this uncertainty as a reference to determine whether the CG passes the Climate Turing Test over the 120 ka to 20 ka test interval.

The CGfamous simulated fields have a smaller RMSE relative to FAMOUS (Temperature: about 50% less and precipitation about 33% less) in the test part (Table 2.4) compared to the CCSM uncertainty (RMSE) listed on (Table 2.3). CGfamous also has relatively better fits over the ice region. CGfamous extracts varying vertical temperature gradients given the significantly reduced misfits over high elevation regions compared to that of the EBM (Figure 2.6). The main comparative deficiciency is the significant dry bias over the Great Lakes (20 ka) and east thereof at 100 ka (Figure 2.7).

The CGfamous simulated temperature field has a cold bias (over the test interval) compared to FAMOUS (February and August) while CGccsm has a cold bias (August) and sharp warm bias (February) compared with FAMOUS over the test interval. A similar pattern of bias occurs over the training interval (Figure 2.5). The FAMOUS output has more variance compared to that of CCSM (Figure 2.5) in part due to the lack of available matched fields (only the radiative surface temperature from the CCSM and the 1.5 meter air temperature from FAMOUS were available).

The imperfection between climate generator predictions and reality can be conceptually broken down into two components. The first is the stochastic process error between the CG and GCM and the second is the structural error of the GCM relative to reality. Our simulated precipitation field has less variance compared to FAMOUS, and future development of the CG will explore other predictor sets which have relevance to precipitation prediction such as hydrology components.

2.8 Conclusions

We have introduced the concept of a Climate Generator to create a large spatio-temporal scale climate representation for coupled ice sheet modelling over glacial cycles. The CG expands the scale of weather generators. For this proof of concept, the CG was implemented over North America. For validation, we compared CGfamous simulated fields against FAMOUS simulated fields (over the test interval which was not used for training the Bayesian artificial neural networks in the CG). We introduced the Climate Turing Test concept to provide a pass/fail reference for field comparison. The FAMOUS GCM was used for CG proof of concept/validation and then the CG was retrained against the much more advanced CCSM (CGccsm). CGfamous and CGcssm have test and training interval errors with respect to their corresponding GCMs that are of the same scale (and mostly less than) our minimal structural error estimate. This estimate is based on the difference between FAMOUS and CCSM temperature and precipitation fields. As such, the CG passes the Climate Turing test. It was not all a priori clear whether this would be possible given the CG reliance on the Energy Balance climate model. The CG will be coupled to the Glacial Systems Model (GSM) for experiments over the last glacial cycle. We expect through the development of our CG, the GSM will be provided with enhanced climate forcing (Temperature and precipitation) relative to previous

experiments. To simulate more atmospheric variables (like evaporation, etc.) the CG needs to be retrained with those GCM fields. In future work, the CG will tested for use with all the major ice-sheets of the last glacial cycle. The CG approach will also be implemented with more advanced EMICs (e.g. LOVECLIM) for shorter time scale contexts (given their increased computational expense).

2.9 Co-authorship statement

M. Arif wrote the initial draft of this manuscript and carried out much of the implementation and testing. T. Hauser started the project implementation. L. Tarasov created the "Climate Generator" and "Climate Turing Test" concepts, oversaw research design, and heavily edited this manuscript.

Bibliography

Pierre Ailliot, Denis Allard, Valérie Monbet, and Philippe Naveau. Stochastic weather generators: an overview of weather type models. *Journal de la Société Française de Statistique*, 156(1):101–113, 2015.

Rasmus E Benestad, Inger Hanssen-Bauer, and Deliang Chen. *Empirical-statistical downscaling*, volume 41. World Scientific, 2008.

JT Chu, J Xia, C-Y Xu, and VP Singh. Statistical downscaling of daily mean temperature, pan evaporation and precipitation for climate change scenarios in haihe river, china. *Theoretical and Applied Climatology*, 99(1-2):149–161, 2010.

Martin Claussen, L Mysak, A Weaver, Michel Crucifix, Thierry Fichefet, M-F Loutre, Shlomo Weber, Joseph Alcamo, Vladimir Alexeev, André Berger, et al. Earth system models of intermediate complexity: closing the gap in the spectrum of climate system models. *Climate Dynamics*, 18(7): 579–586, 2002.

G Deblonde, WR Peltier, and WT Hyde. Simulations of continental ice sheet growth over the last glacial-interglacial cycle: experiments with a one level seasonal energy balance model including seasonal ice albedo feedback. *Global and planetary change*, 6(1):37–55, 1992.

Yonas B Dibike and Paulin Coulibaly. Temporal neural networks for downscaling climate variability and extremes. *Neural Networks*, 19(2):135–144, 2006.

Hayley J Fowler and Rob L Wilby. Beyond the downscaling comparison study. *International Journal of Climatology*, 27(12):1543–1545, 2007.

Hugues Goosse, V Brovkin, T Fichefet, R Haarsma, P Huybrechts, J Jongma, Anne Mouchet, F Selten, P-Y Barriat, J-M Campin, et al. Description of the earth system model of intermediate complexity loveclim version 1.2. *Geoscientific Model Development*, 3:603–633, 2010.

Tristan Hauser and Entcho Demirov. Development of a stochastic weather generator for the subpolar north atlantic. *Stochastic Environmental Research and Risk Assessment*, 27(7):1533–1551, 2013.

Tristan Hauser, Andrew Keats, and Lev Tarasov. Artificial neural network assisted bayesian calibration of climate models. *Climate dynamics*, 39(1-2):137–154, 2012.

William T Hyde, Kwang-Yul Kim, Thomas J Crowley, and Gerald R North. On the relation between polar continentality and climate: Studies with a nonlinear seasonal energy balance model. *Journal of Geophysical Research: Atmospheres*, 95(D11):18653–18668, 1990.

Sik-Yum Lee. *Structural equation modeling: A Bayesian approach*, volume 711. John Wiley & Sons, 2007.

Z Liu, BL Otto-Bliesner, F He, EC Brady, R Tomas, PU Clark, AE Carlson, J Lynch-Stieglitz, W Curry, E Brook, et al. Transient simulation of last deglaciation with a new mechanism for bølling-allerød warming. *Science*, 325(5938):310–314, 2009.

L Loulergue et al. Epica dome c ice core 800kyr methane data. *IGBP PAGES/World Data Center for Paleoclimatology Data Contribution Series*, 54, 2008.

Y Lu and XS Qin. A coupled k-nearest neighbour and bayesian neural network model for daily rainfall downscaling. *International Journal of Climatology*, 34(11):3221–3236, 2014.

Qunying Luo, Li Wen, John L McGregor, and Bertrand Timbal. A comparison of downscaling techniques in the projection of local climate change and wheat yields. *Climatic change*, 120(1-2): 249–261, 2013.

D Luthi et al. Epica dome c ice core 800kyr carbon dioxide data. *IGBP PAGES/World Data Center for Paleoclimatology Data Contribution Series*, 55, 2008.

Saumen Maiti, Gautam Gupta, Vinit C Erram, and Ram Krishna Tiwari. Delineation of shallow resistivity structure around malvan, konkan region, maharashtra by neural network inversion using vertical electrical sounding measurements. *Environmental earth sciences*, 68(3):779–794, 2013.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

David B Reusch and Richard B Alley. A 15-year west antarctic climatology from six automatic weather station temperature and pressure records. *Journal of Geophysical Research: Atmospheres (1984–2012)*, 109(D4), 2004.

Justin T Schoof. Statistical downscaling in climatology. *Geography Compass*, 7(4):249–265, 2013.

Justin T Schoof and SC Pryor. Downscaling temperature and precipitation: A comparison of regression-based methods and artificial neural networks. *International Journal of Climatology*, 21 (7):773–790, 2001.

Robin S Smith and Jonathan Gregory. The last glacial cycle: transient simulations with an aogcm. *Climate dynamics*, 38(7-8):1545–1559, 2012.

L. Tarasov and W. R. Peltier. Terminating the 100 kyr ice age cycle. *jgr*, 102(D18):21665–21693, 1997.

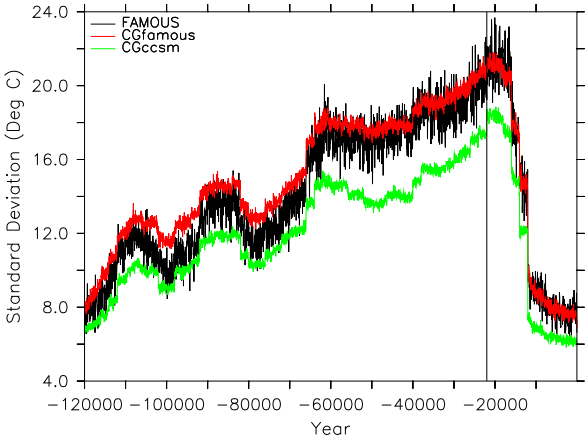
Lev Tarasov, Arthur S Dyke, Radford M Neal, and W Richard Peltier. A data-calibrated distribution of deglacial chronologies for the north american ice complex from glaciological modeling. *Earth and Planetary Science Letters*, 315:30–40, 2012.

RL Wilby, SP Charles, E Zorita, B Timbal, P Whetton, and LO Mearns. Guidelines for use of climate scenarios developed from statistical downscaling methods. 2004.

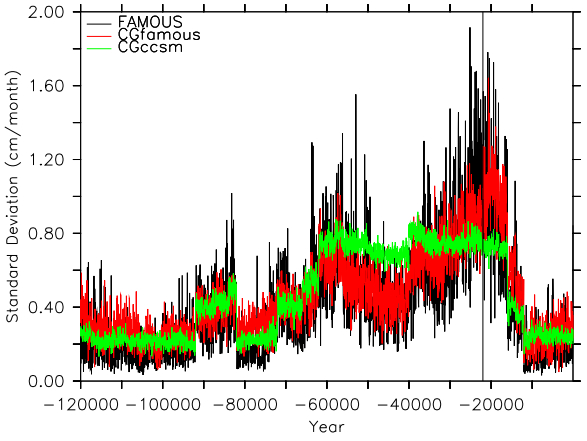
Appendix A

Supplement of Climate Generator

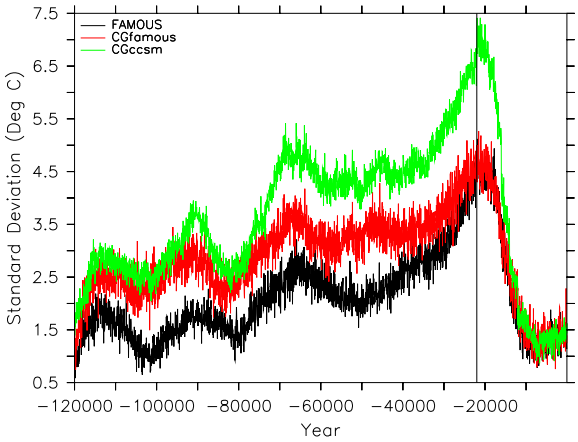
This appendix contains time series and map plots comparison of CGfamous and CGccsm simulated fields with FAMOUS and EBM climate model outputs.



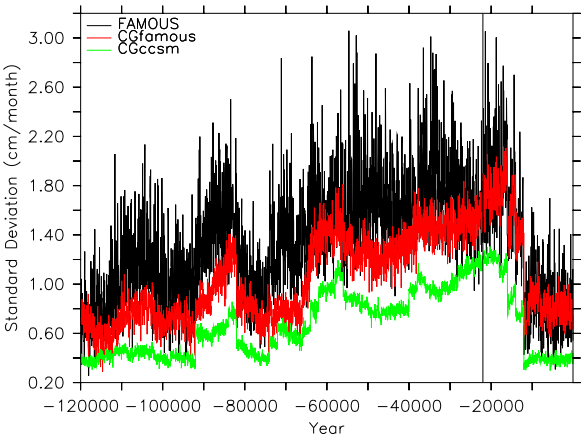
(a) February Temperature



(b) February Precipitation

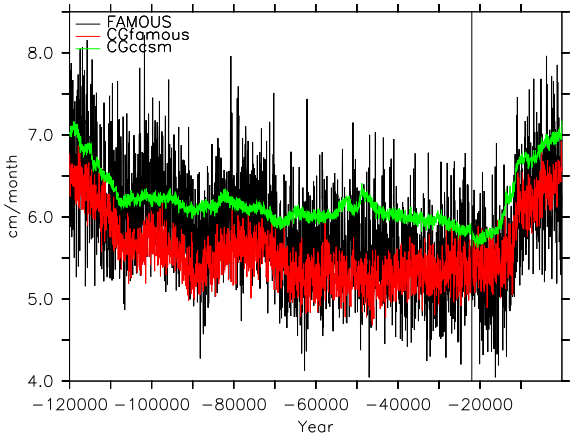


(c) August Temperature

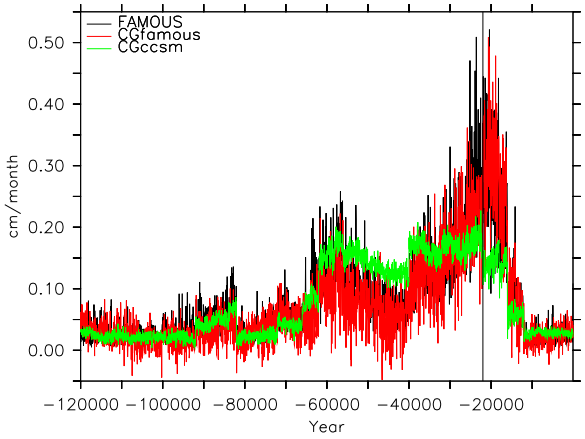


(d) August precipitation

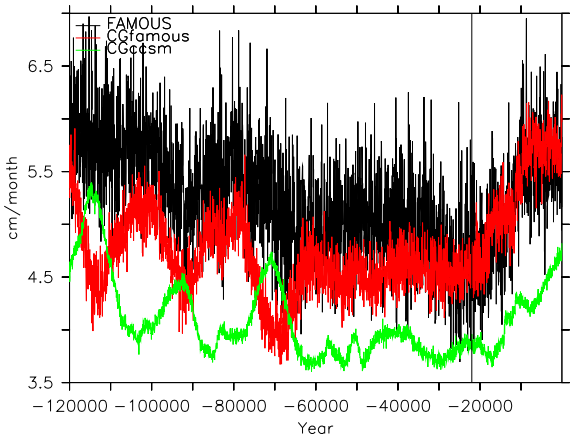
Figure A.1: Comparison of the spatial variance (standard deviation) on the ice region. The black vertical line separates test (left) and training part (right).



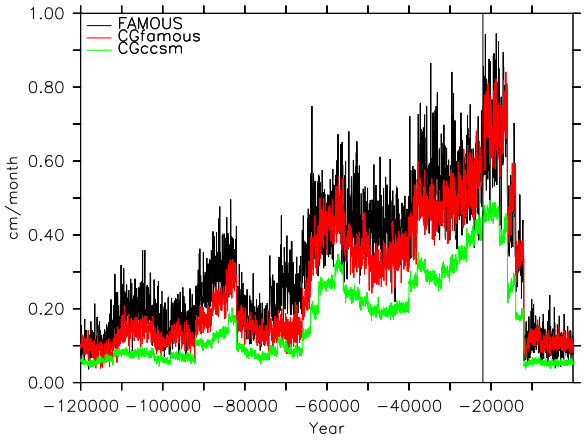
(a) February (Full grid area)



(b) February (Ice region)



(c) August (Full grid area)



(d) August (Ice region)

Figure A.2: Comparison of the spatial mean (with latitudinal weighting) precipitation time series. The black vertical line separates test (left) and training part (right).

Figure A.3: February temperature field (Deg C) at 18 ka with the elevation and ice contour shown in black and blue. Difference between plots are shown in 3rd row. Model names and months are indicated in the top left corner in each box).

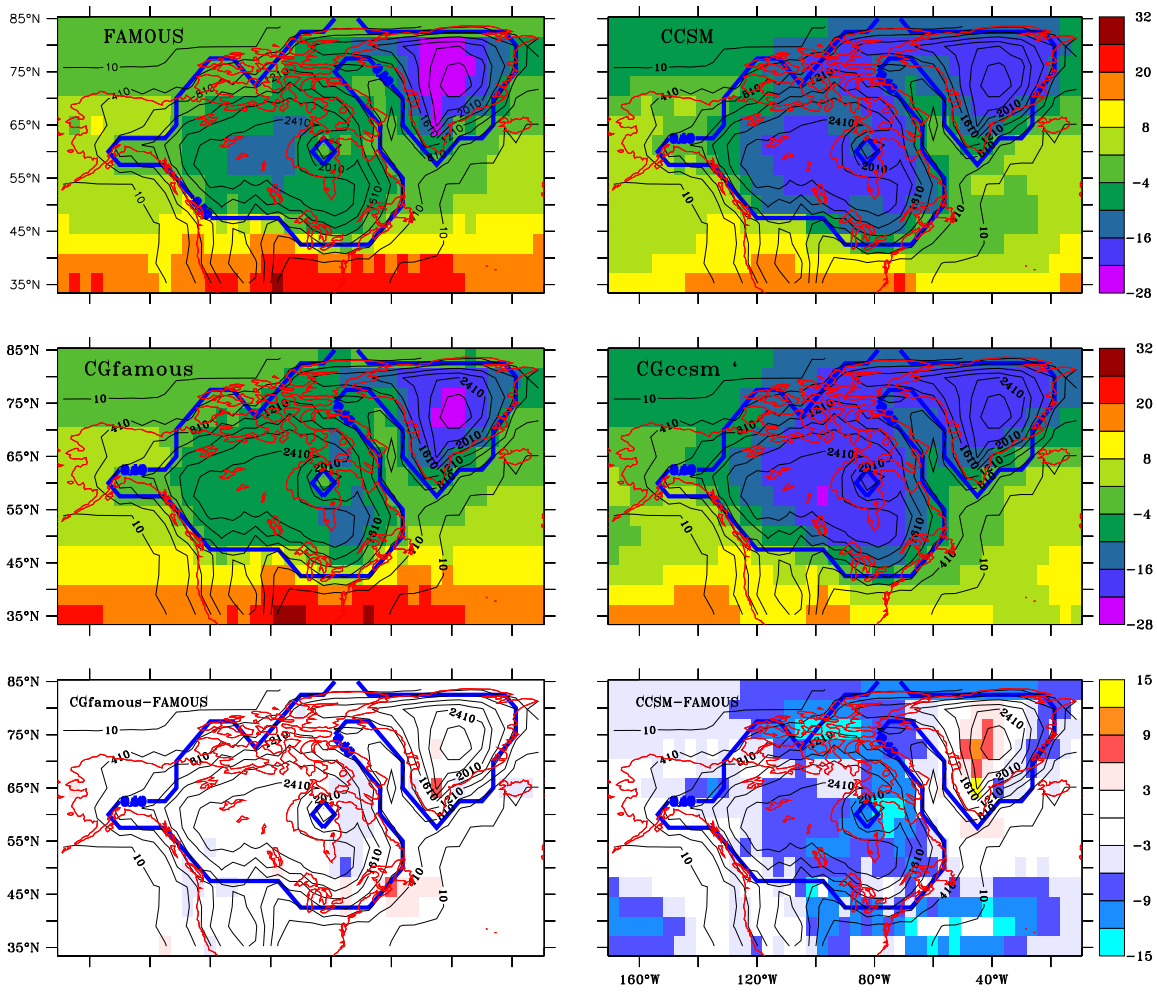


Figure A.4: August temperature field (Deg C) at 18 ka with the elevation and ice contour shown in black and blue. Difference between plots are shown in 3rd row. Model names and months are indicated in the top left corner in each box).

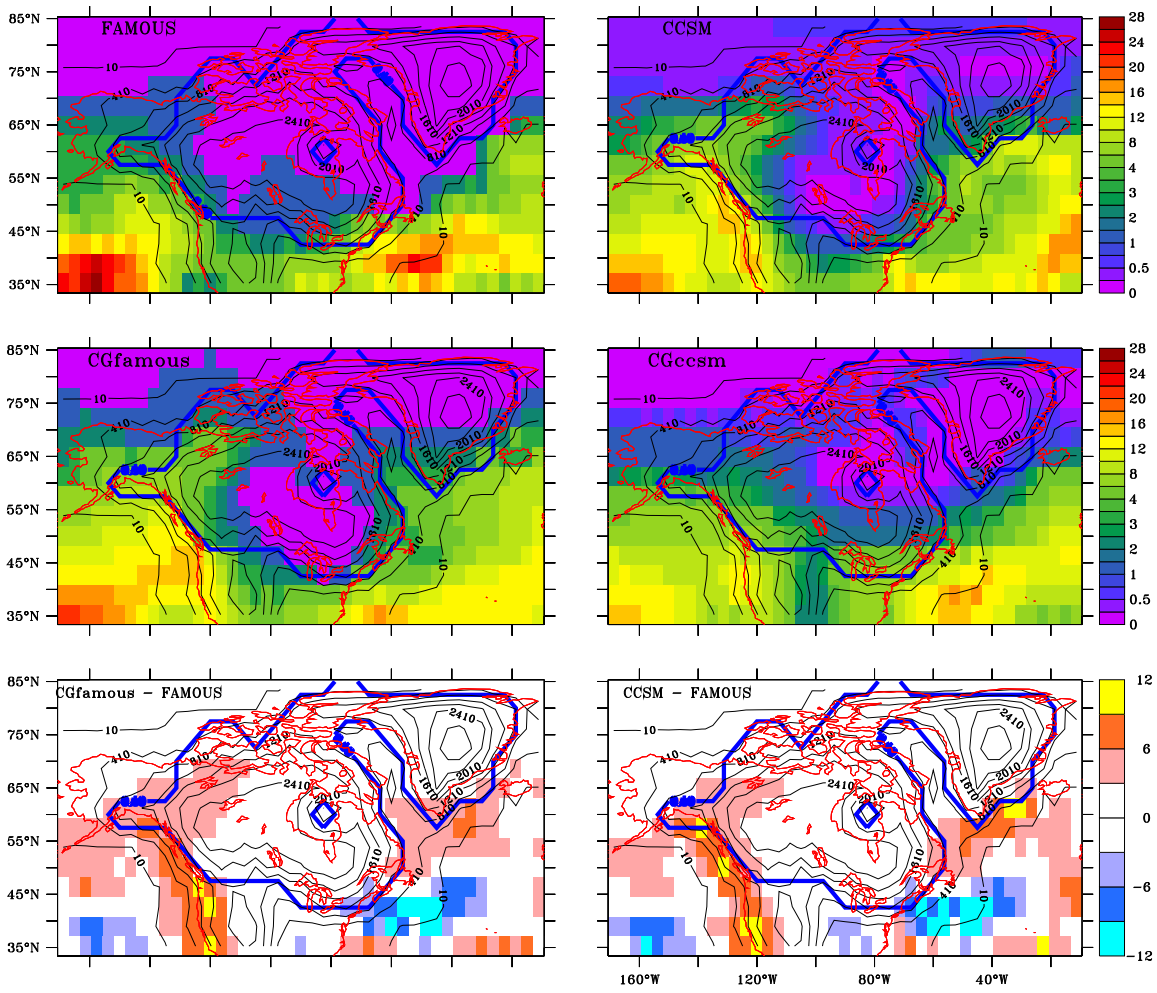


Figure A.5: February precipitation field (cm/month) at 18 ka with the elevation and ice contour shown in black and blue. Models name and month are indicated in the top left corner in each box. Difference between plots are shown in 3rd row.

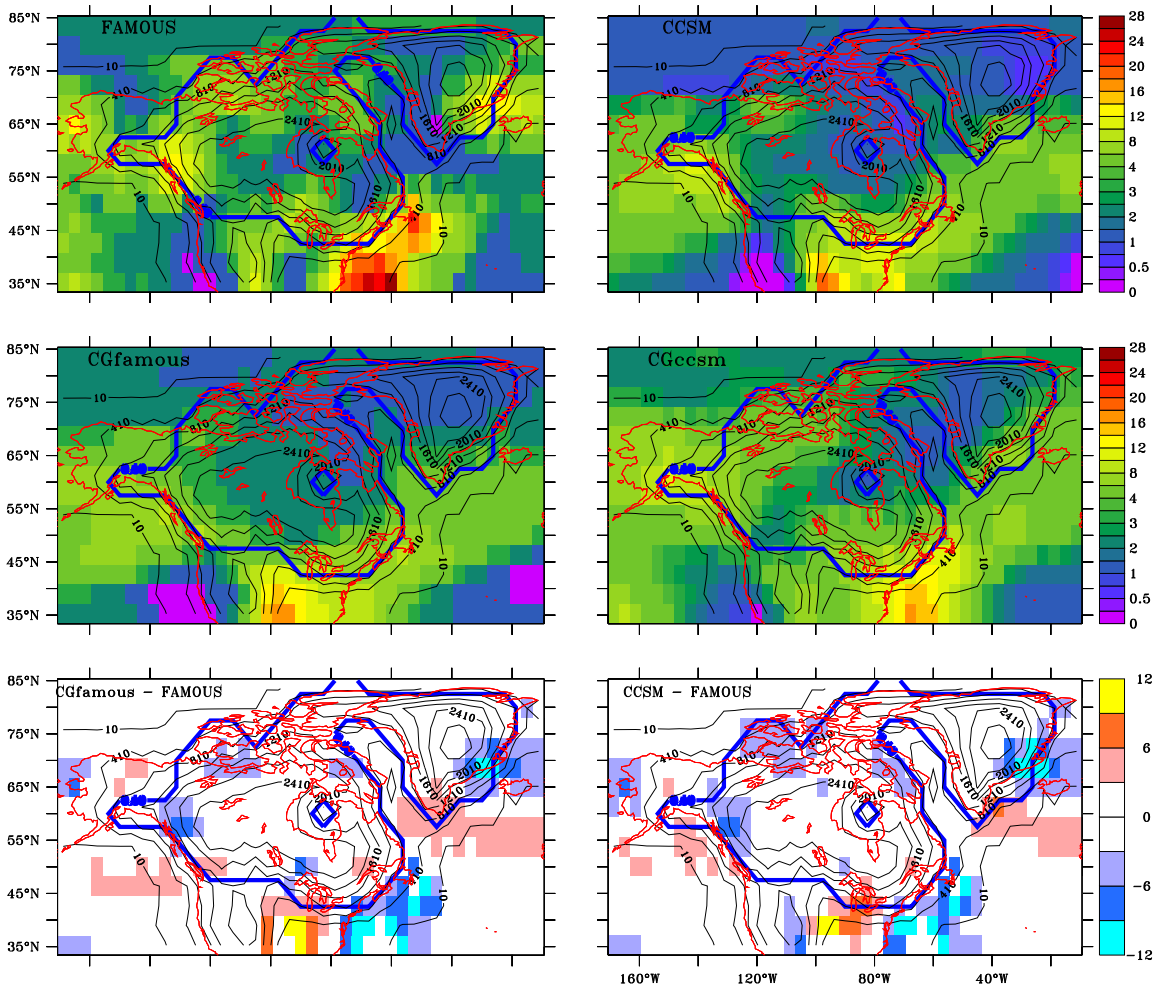


Figure A.6: August precipitation field (cm/month) at 18 ka with the elevation and ice contour shown in black and blue. Difference between plots are shown in 3rd row. Model names and month are indicated in the top left corner in each box).

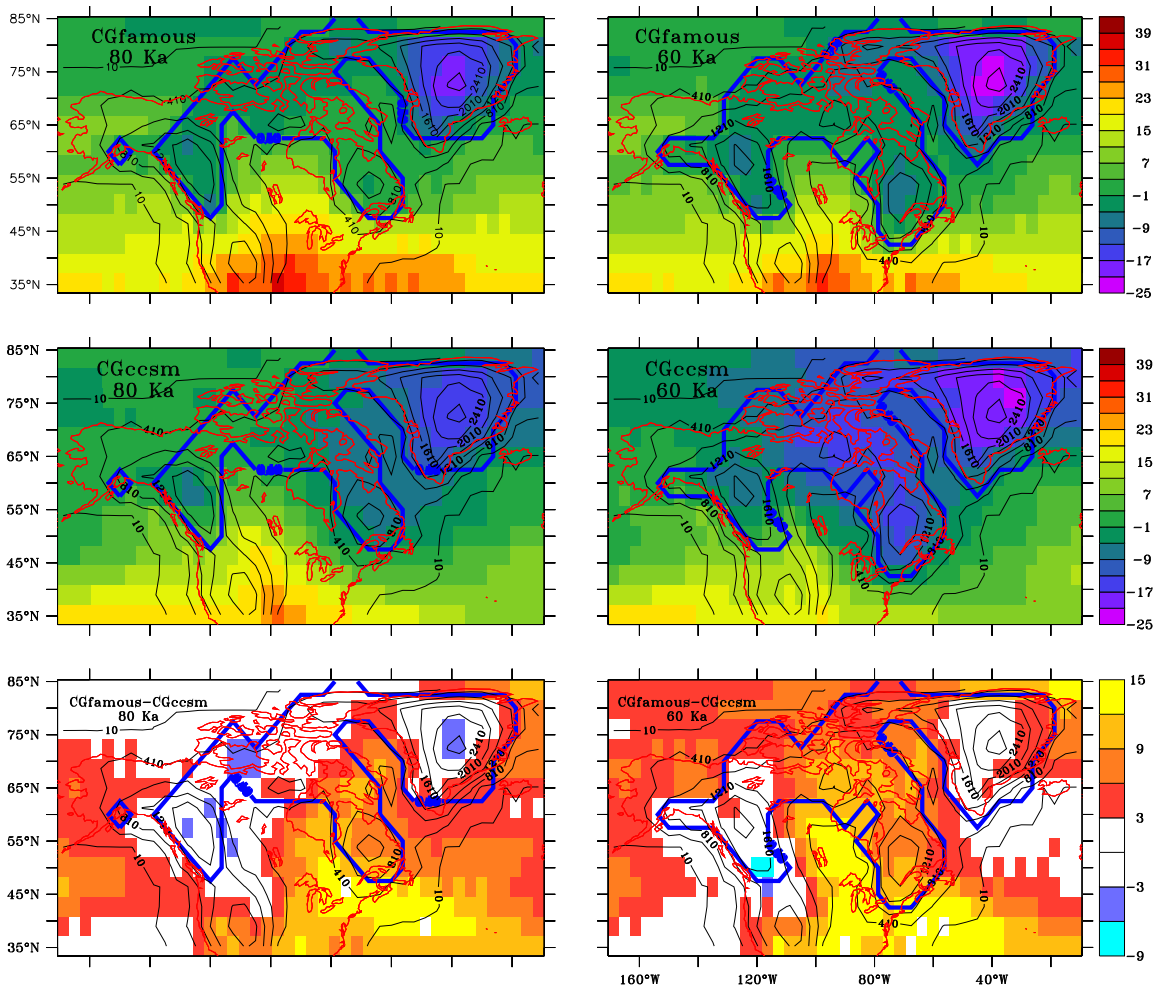


Figure A.7: The August temperature field (Deg C). (1st and 2nd row) with the elevation and ice contour shown in black and blue. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box).

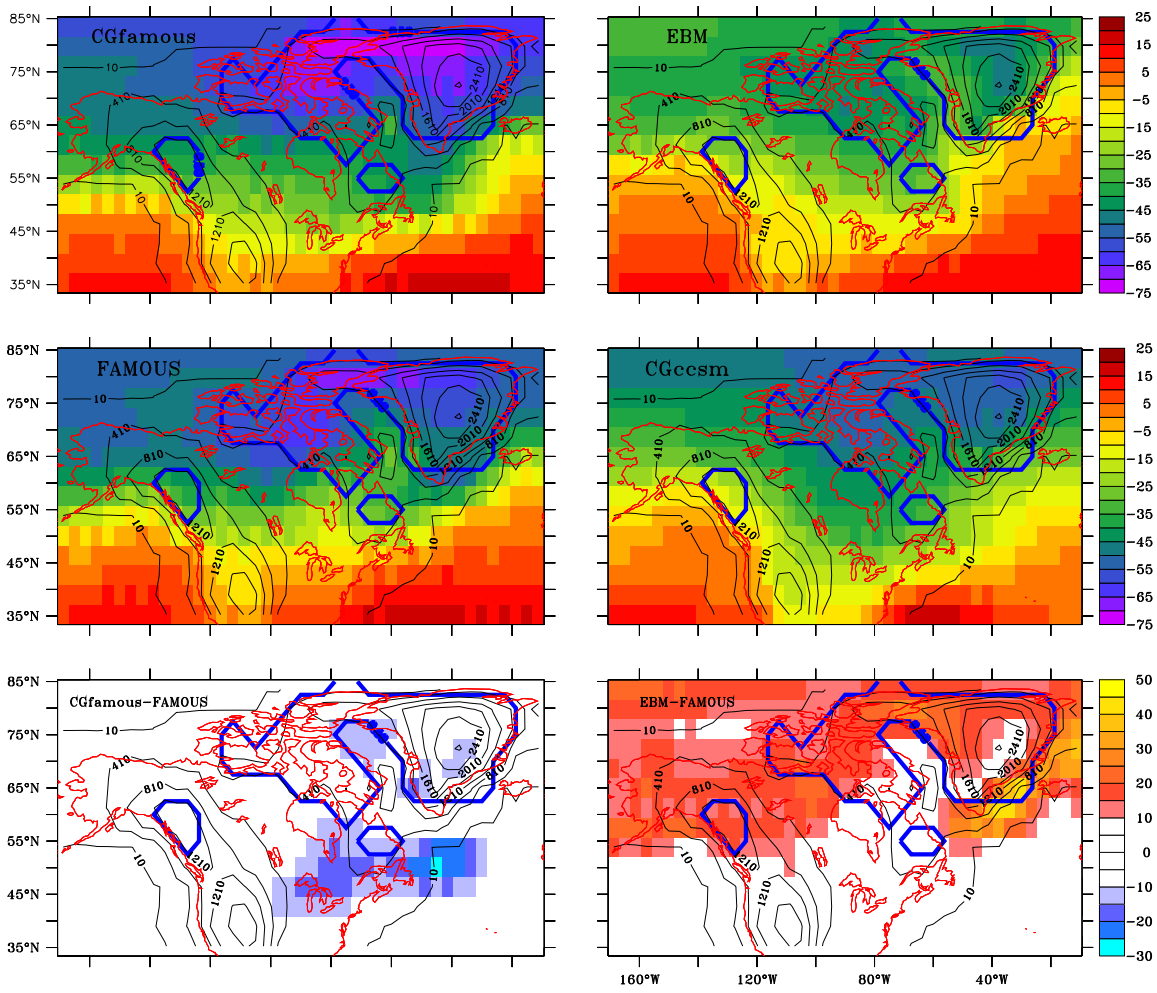


Figure A.8: The February temperature field (Deg C) at 100 ka (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box.

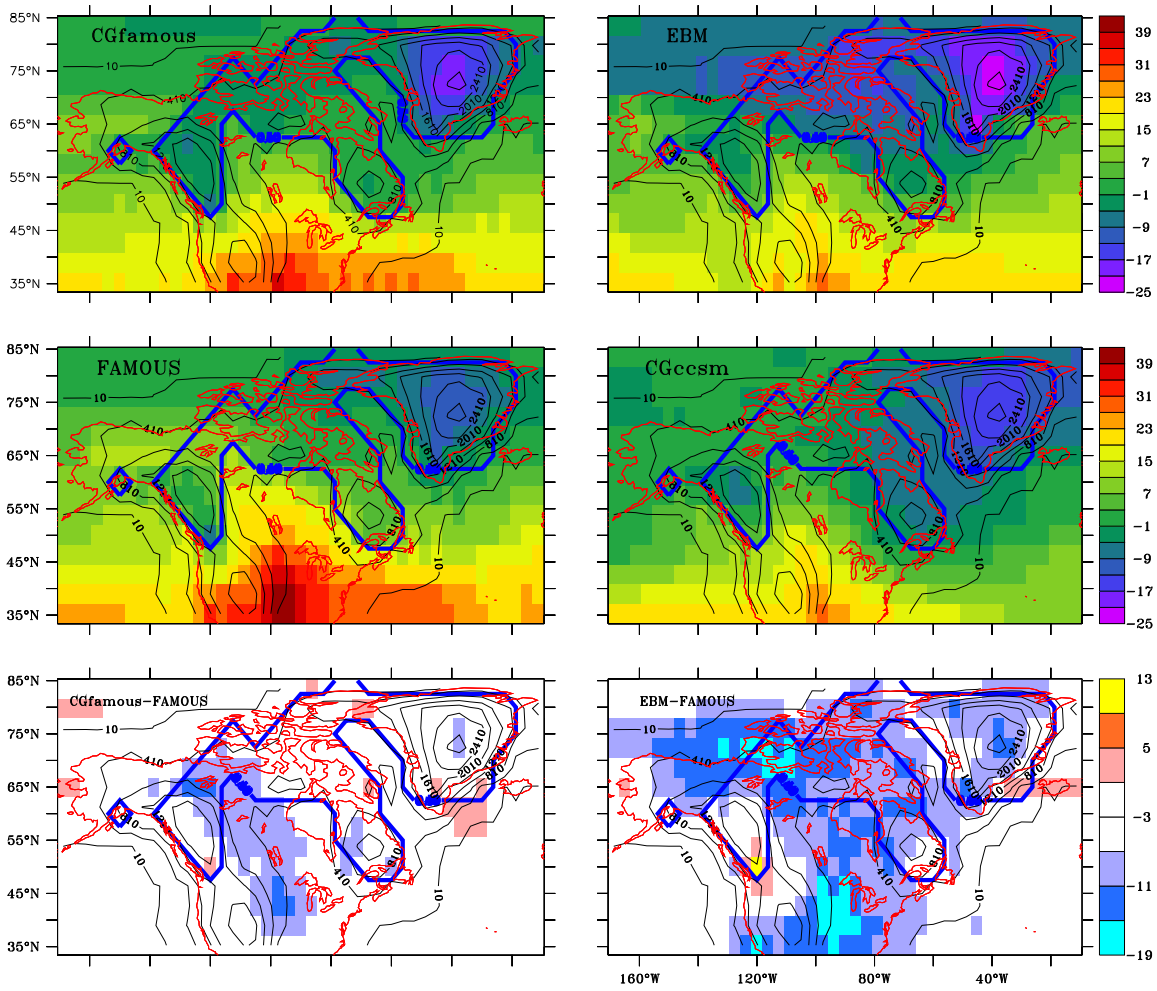


Figure A.9: The August temperature field (Deg C) at 80 ka (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box.

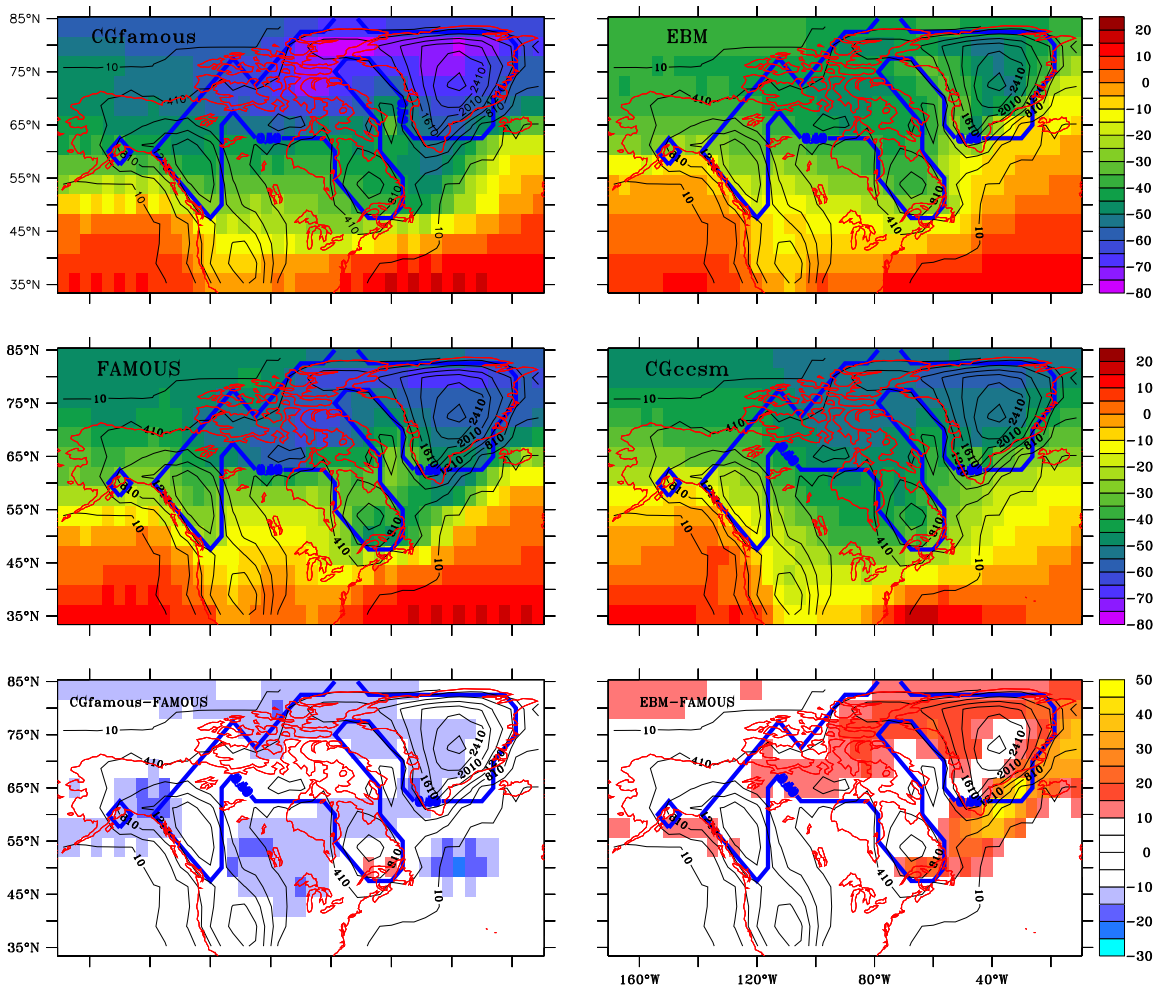


Figure A.10: The February temperature field (Deg C) at 80 ka (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box.

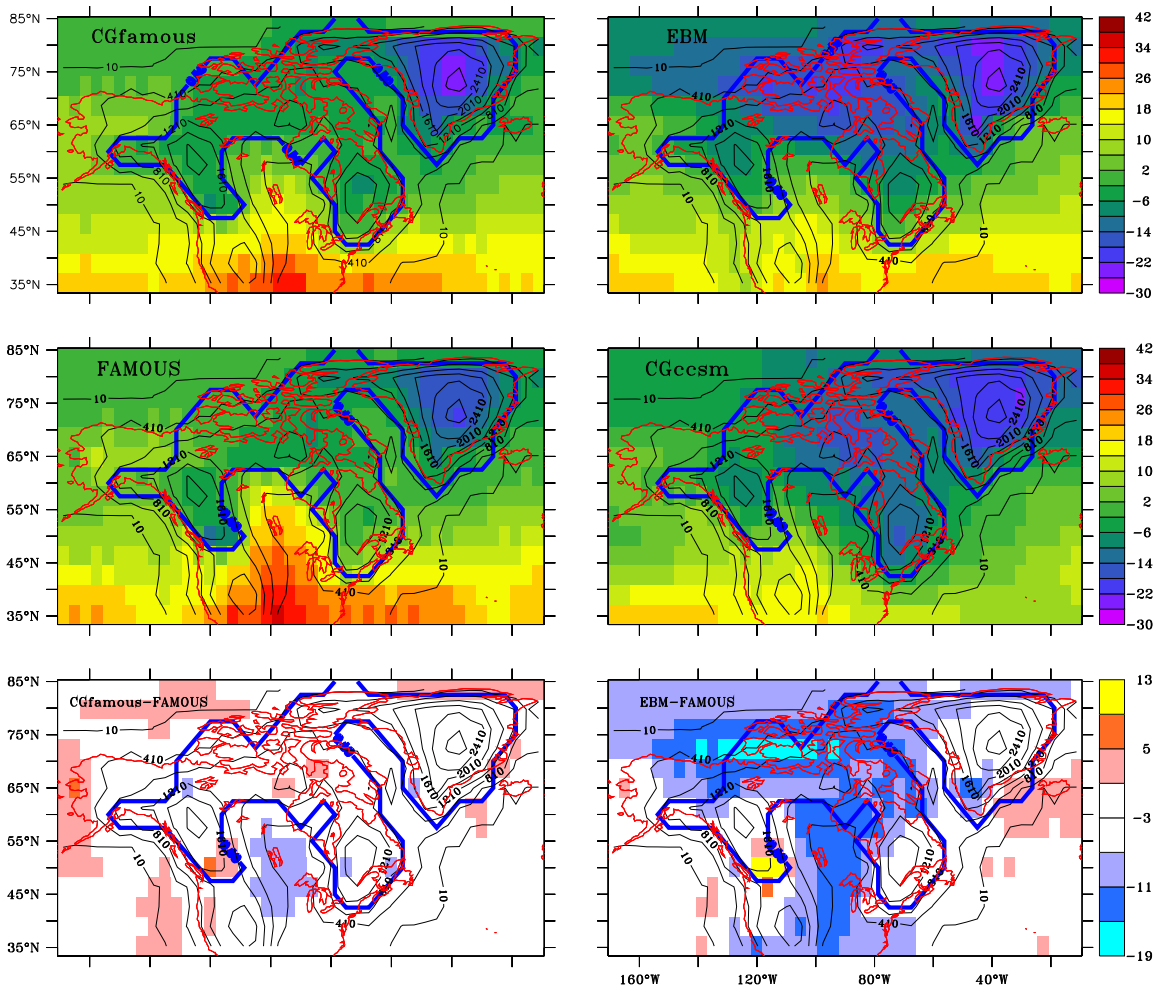


Figure A.11: The August temperature field at 60 ka (Deg C) (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box.

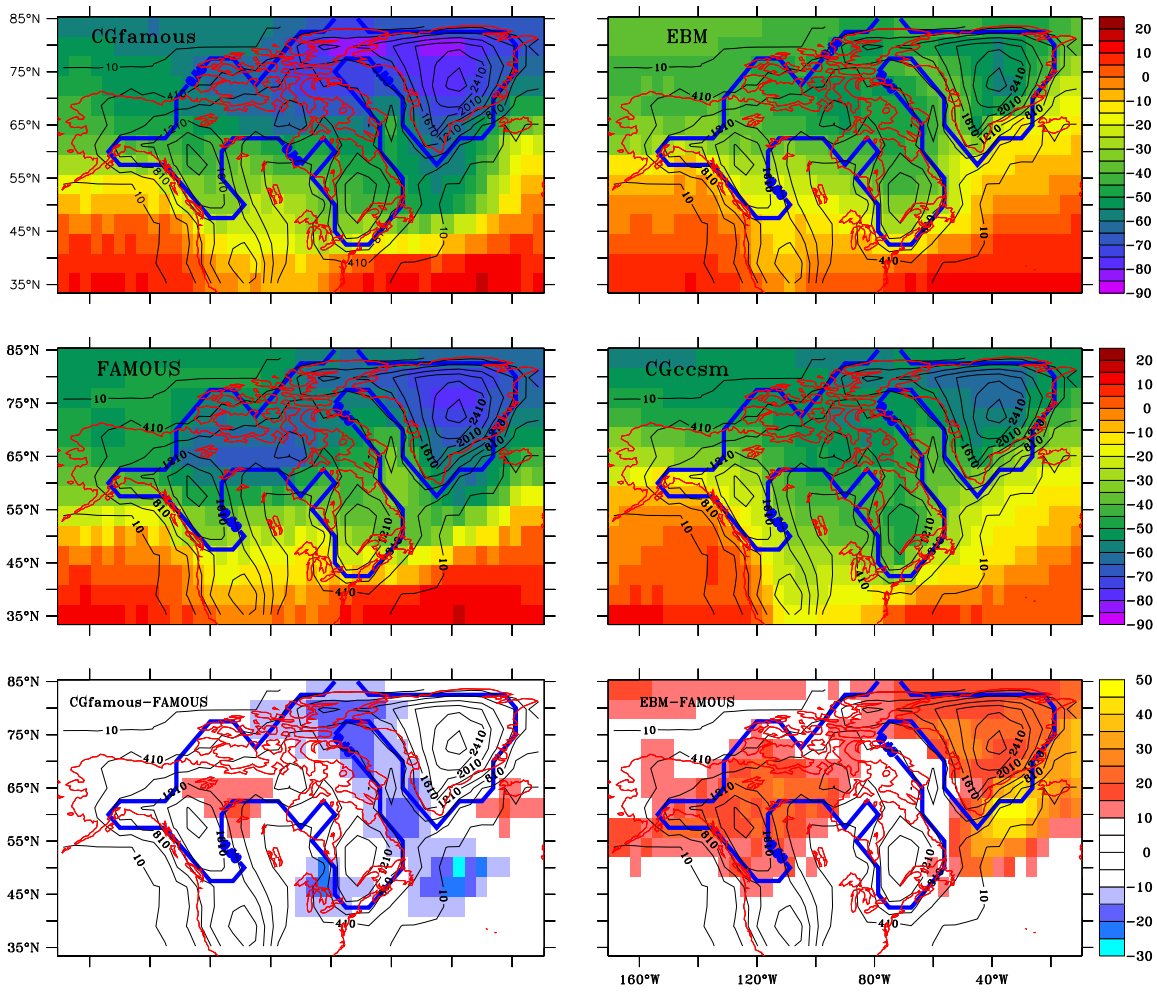


Figure A.12: The February temperature field (Deg C) at 60 ka (1st and 2nd row) with the elevation and ice contour shown in black and blue respectively. Difference between plots are shown in 3rd row. Model names are indicated in the top left corner in each box.

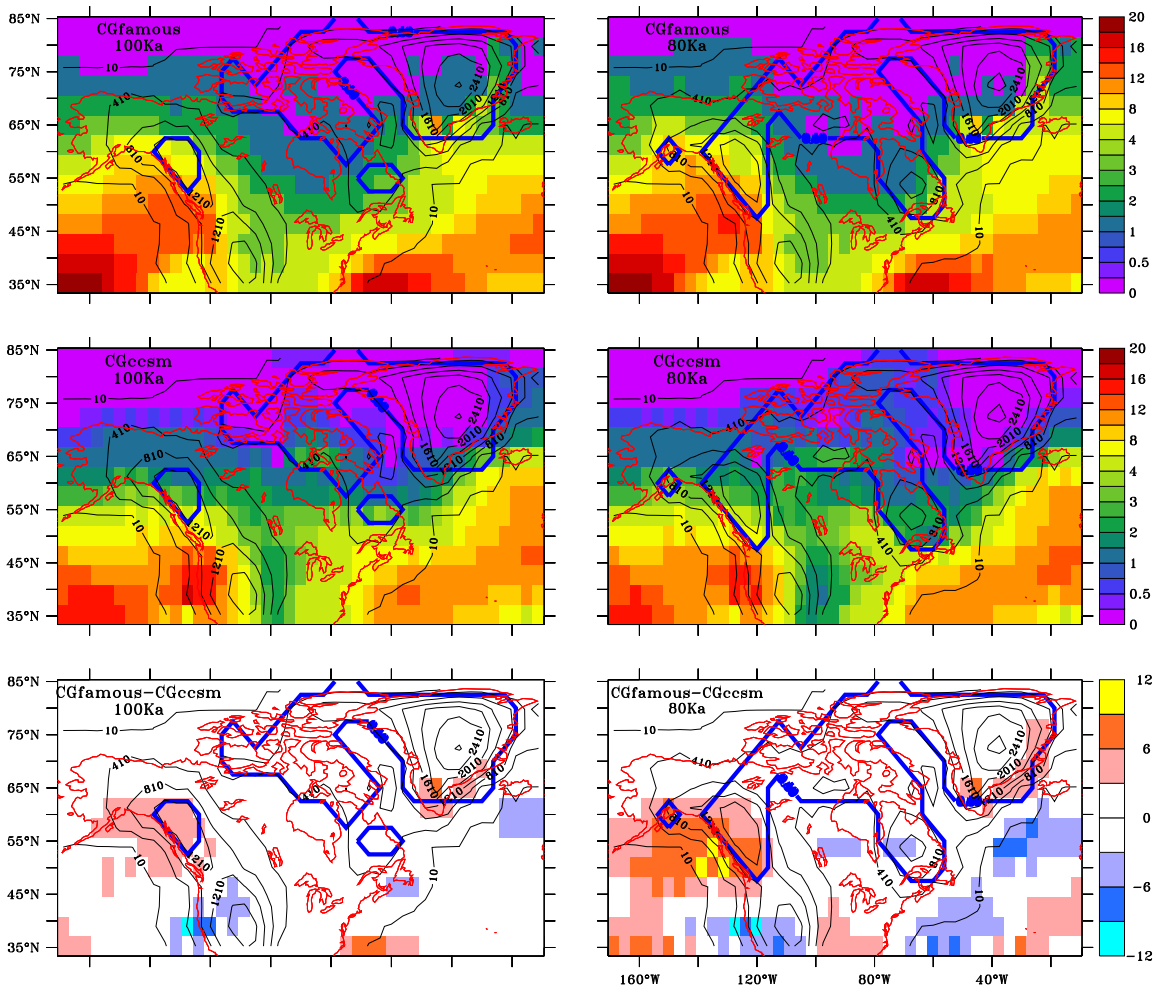


Figure A.13: February precipitation field (cm/month). Models name and times indicated in the top left corner in each box.

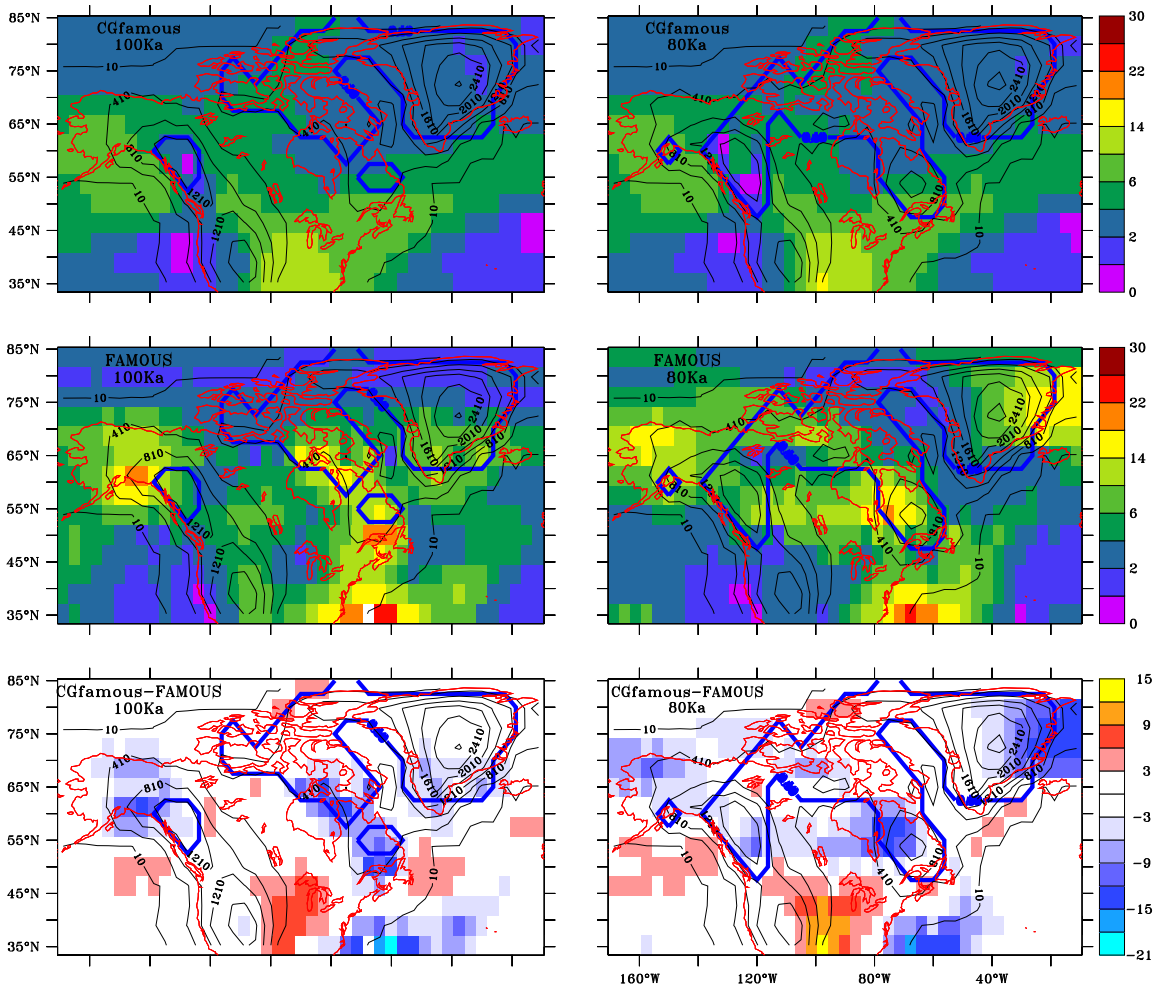


Figure A.14: August precipitation field (cm/month) (1st and 2nd row). Difference between plots are shown in 3rd row. Model names and times are indicated in the top left corner in each box.

Correlation map between residual (ie CG without noise - FAMOUS) prec and temperature

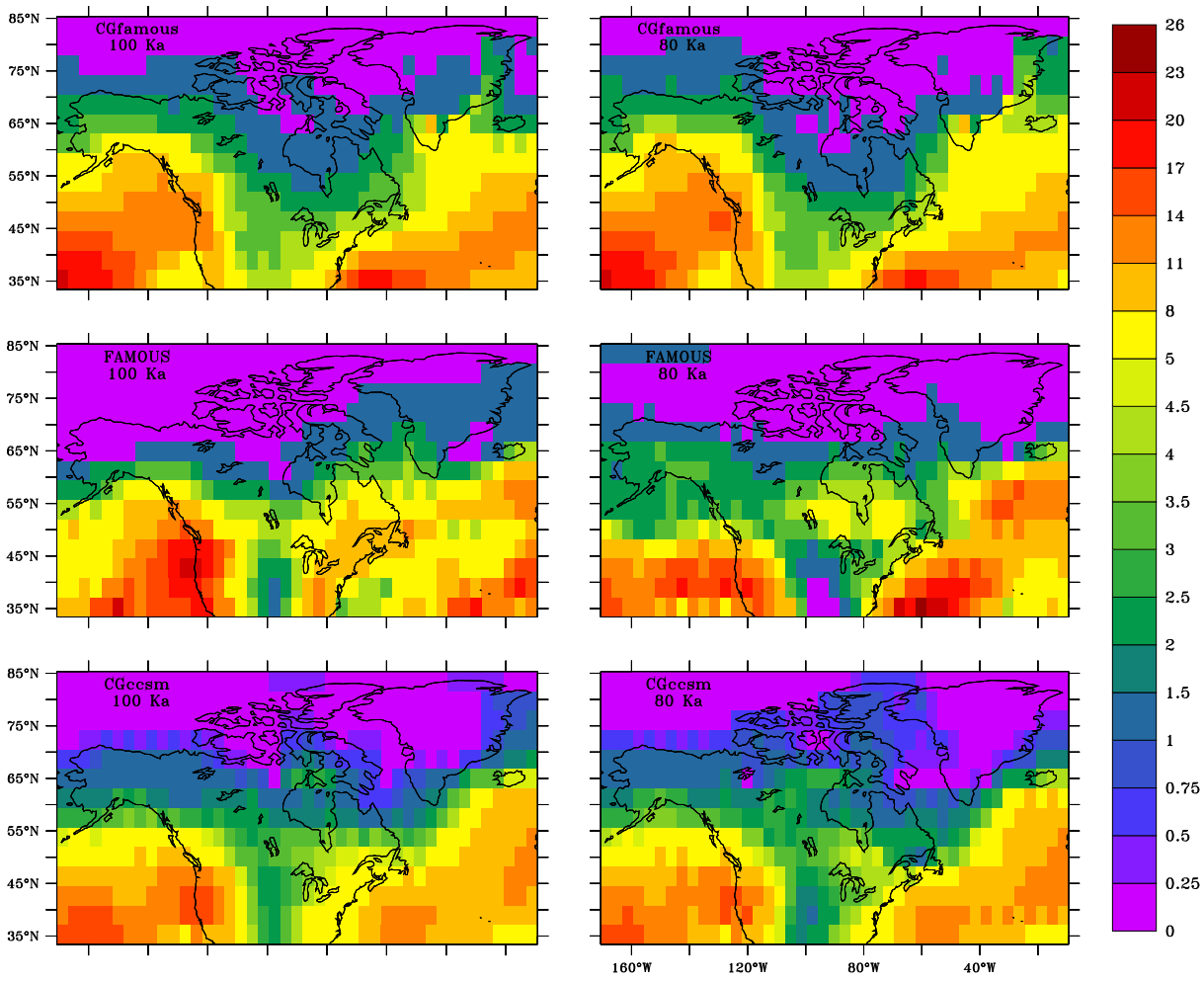


Figure A.15: The February precipitation field (cm/month). Model names and times are indicated in the top left corner in each box.

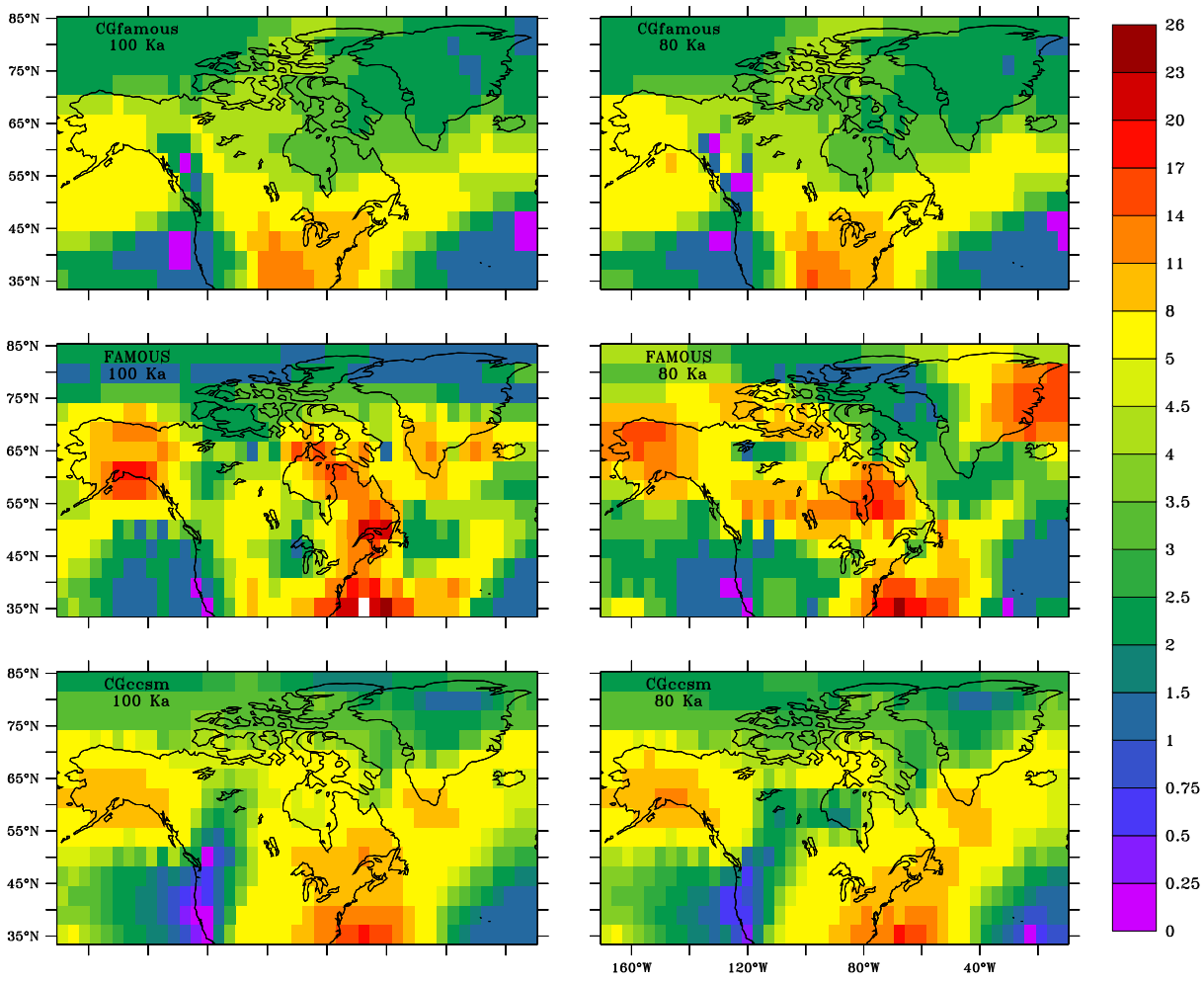


Figure A.16: The August precipitation field (cm/month). Model names and times are indicated in the top left corner in each box.

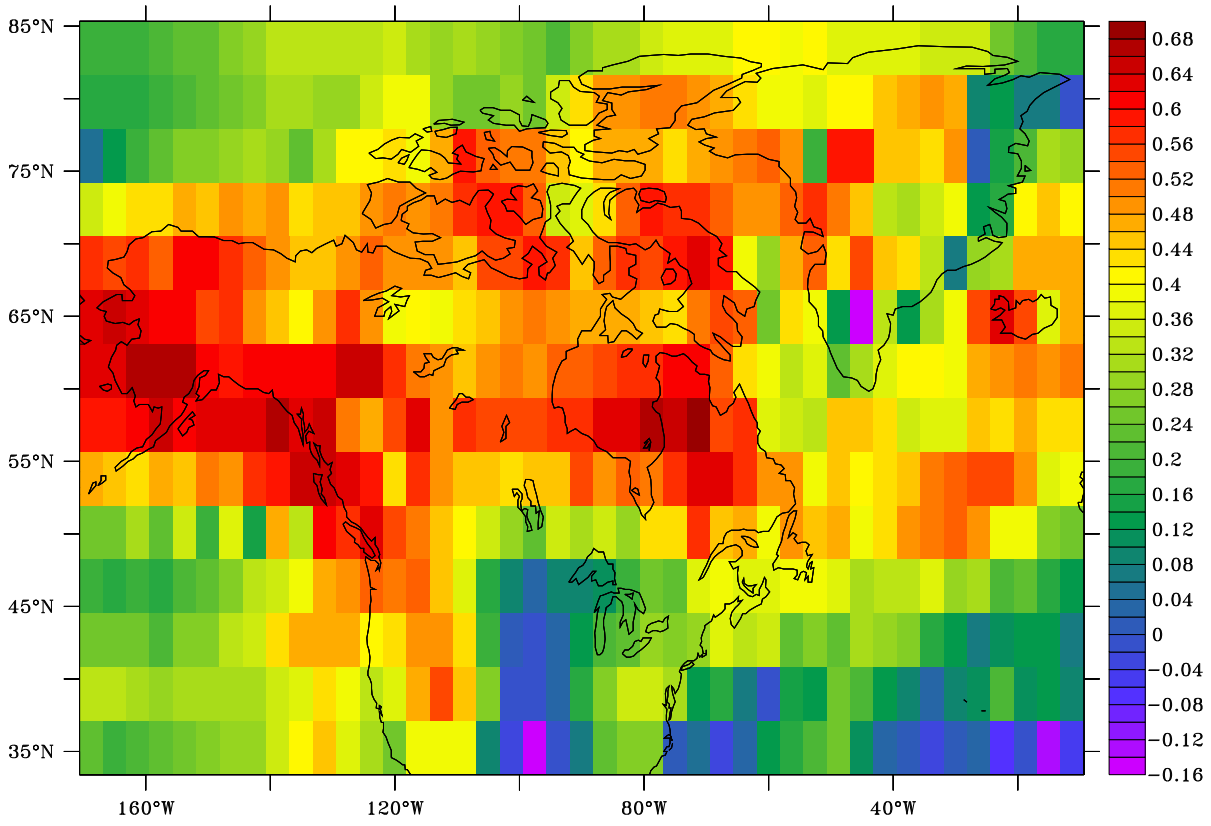


Figure A.17: February correlation map between residual (i.e. CG without noise - FAMOUS) Temperature and precipitation fields.

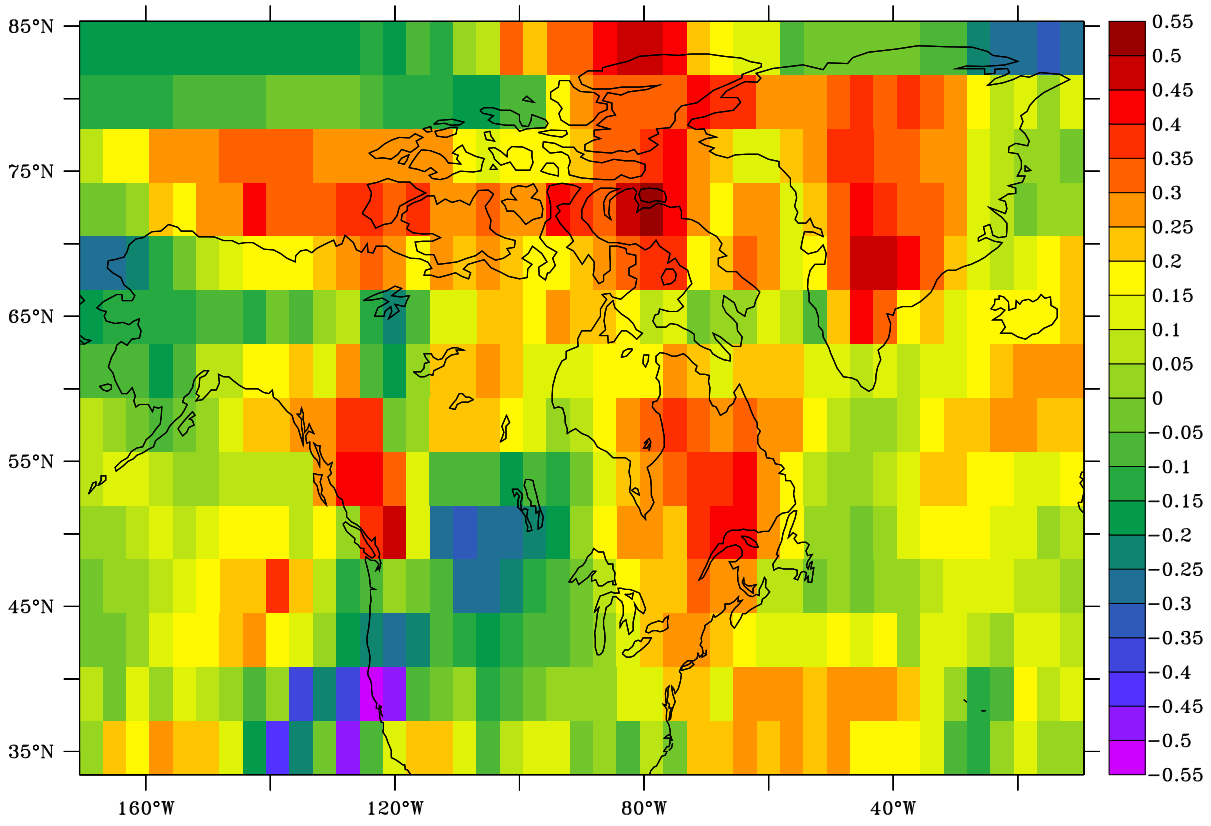


Figure A.18: August correlation map between residual (i.e. CG without noise - FAMOUS) Temperature and precipitation fields.