Supplementary online information for **Exploratory predictive likelihood for real chaotic systems: Predictors for local rainfall and temperature**

Detailed Background

The essential statistical object in embedding is the delay map. A delay map[1,2] has the the following form. If average seasonal temperature, T_i, for season i, and total seasonal precipitation, P_i, for a given location are variables in the delay map, one element of the delay map may be $(T_i, P_{i-2}, T_{i-2}, T_{i-5})$, and the immediately preceding element $(T_{i-1}, P_{i-3}, T_{i-3}, T_{i-6})$. This delay map has dimension 4. the set of delay maps with dimension larger than twice the box dimension of the attractor, (box dimension is the limit of the ratio of the number of boxes (hypercubes) that cover the dimension in phase space to the negative log of the length of the side of the box as that length goes to 0) , that are diffeomorphic to the original attractor are “prevalent” [2,3] which is an infinite dimensional version of almost sure. For example if the box dimension is 1.4, then the delay map above fits the criterion, and delay maps of this dimension that are diffeomorphic are prevalent. To see how this could be used in prediction, suppose we wish to predict temperature 2 seasons ahead at a particular geographical point. Then, a delay map where the size is large enough and we have a temperature measurement that is 2 seasons in the future of the rest of the data is likely diffeomorphic to the original attractor (as in the example delay map). So where the attractor has a differentiable manifold, each point along such delay maps has a tangent plane. To estimate the dimension of the manifold, a method like that of Bickel and Levina[4] can be used. Because of the ergodicity of the attractor (Eckerd and Ruelle[5]) if n is the number of measurements and we take a shrinking neighborhood of the point we are trying to predict at (say at rate $n^{-\delta}, \delta < 1, \text{Stone}, [6]$) then by Neumann’s [7] result as long as the correlation of the measurements is weak over time we can assume that we can estimate the local linear regressions and local density accurately using standard methods . Ruelle [8], has shown that diffeormorphisms of axiom A attractors show strong mixing exponentially fast implying Neumanns weak correlation for the dynamic system. Assuming an extended version of the chaoticity hypothesis [9,10] we can therefore get consistent predictions of the future using local linear regression on almost any delay map of sufficient size, and we can get consistent density estimators using data from appropriately shrinking neighborhoods based on multiview embeddings[11,12].

Suppose measurement error is non negligible. Lalley [13] provides a definition of nonnegligible, and Judd [14] shows how standard statistical methods may fail. In this case, unless the measurement error is highly constrained, (conditions in Lalley, ibid) the trajectory itself is not uniquely identifiable, which necessarily means that predictions based on statistical models can not be uniquely identifiable, and in fact prediction through a dynamic model adjusted to agree on some measured “ initial conditions” input from observables would also not be unique.

Systems that are open, so that phenomena outside the system can effect them will have a similar issue. Small perturbations to the dynamics with a differentiable flow has us comparing $f^n(x)“ + \varepsilon$ with measurement error vs $f(f^{n-1}(x) + \varepsilon) \approx f^n(x) + f'(x)\varepsilon$ with perturbations. So the difference between measurement error and small perturbations may be unidentifiable if ε has a heterogeneous variance. If we assume the measurement error (or the random perturbations or both) are ergodic and conditionally independent of the dynamic system, then Nueumann's condition holds for the observations. So multiview embedding will again result in a consistent estimate of the local regression and the density estimates based on them and observed nearest neighbor residuals. So we can estimate an asymptotically consistent predictive distribution, a convolution of the error/perturbation distribution with the SRB measure of the dynamic process, which can be combined with observations of the predicted values to make predictive likelihoods.

Assuming the combined distribution has a density, multiview embedding [ibid] provides an approach to calculate that predictive distribution with the system using a density estimate based on single nearest neighbor for each delay map used in prediction. The predictive density can then provide a basis for a marginal predictive likelihood measure (The observations of the predicted values may not be independent observations so the correlation in the sum of the likelihood terms must be accounted for, or they must be taken far enough apart so independence may be a reasonable assumption).

With the addition of error, either measurement error, or perturbations of the system, the informational equivalence of different delay maps disappears, and we can identify more or less informative variables to use in prediction.

The calculation of the predictive density and the pseudo likelihood proceeds as follows:

Begin with a training data set and a test data set. The training data can be constructed from one or many runs of a climate model, or from historical data. The test data is from historical data. Construct a random set of delay maps. The delay maps are constructed so that the variable to be predicted is the represented with the latest index in time, forward by the desired lag from all the other variables. Denote each element of the delay map as (X,Y) where X is the vector of all other terms in the delay map and Y is the term being predicted. For each element of the test sample, chose a neighborhood around X and identify all elements of the training sample in that neighborhood. Use the (X,Y) pairs from the neighborhood to construct a linear regression of Y on X (I use the LARS[15] algorithm minimizing Cp to choose the regression). Take the prediction as the prediction of that linear regression using the original X from the test sample. For the rest of the calculation there are three possibilities

1) *Take the residual, as the residual of the nearest neighbor (in the X space) from the training sample, where it is taken from the regression on the corresponding X in the training sample. Repeat this over the full set of random delay maps for this one term in the test sample and construct a density* $f(Y)$ *by nonparametric density estimation over the set of* $Prediction + residual\ from\ nearest\ neighbor$ *for all the delay maps*
2) *Proceed as in 1, only bootstrap over the training sample in the neighboorhood, making* B_n *samples for each delay map*
3) *Proceed as in 1, only take the k nearest neighbors instead of just the 1st.*

The results of Neumann[ibid] indicate that any of these should provide consistent estimates of the density $f(Y)$ at this point I am using option 1, but working on 2 to improve performance. The log likelihood at that point is then simply that density evaluated at the observed value of Y in the test sample. The pseudo marginal log likelihood is the sum of the log likelihoods calculated over the full test sample. This is a pseudo marginal log likelihood because it is not perfectly clear that the Y in the test samples can be considered independent. To test for differences in likelihood, we calculate the pairwise differences between the loglikelihoods at each point and construct a t statistic accounting for the significant auto correlations.

The nonparametric density estimates in R have a slower convergence rate (on an already slow convergence rate due to the correlation in the system and the need for whitening by windows to be taking place) so a faster converging but biased density estimate can also be calculated using the EM algorithm [16] to fit a mixture of up to k (k fixed) normal distribution. All results presented in the paper use the kernal estimators.

1) F. Takens, "Detecting Strange Attractors in Turbulence", Lecture notes in mathematics number 898, (Springer Verlag, 1981)
2) Sauer, T., Yoreck, J. and Casdagli, M., 1991 "Embedology", Journal of Statistical Physics, 65, 579-616
3) Hunt, Sauer, York, 1992, "Prevalence, a translation invariant "almost every" on infinite dimensional spaces", Bulletin of the AMS, 1992, v27, n2, October 1992, 217-238.
4) Levina, E. and Bickel, P. J. (2005). "Maximum likelihood estimation of intrinsic dimension". Advances in NIPS 17. MIT Press
5) Eckmann, J.P., and Ruelle, D.(1985), "Ergodic Theory of chaos and strange attractors", Reviews of modern Physics, 57, 617-656, (1985)
6) Charles J. Stone, ***The Annals of Statistics***, Vol. 5, No. 4 (Jul., 1977), pp. 595-620 (26 pages)
    a. https://www.jstor.org/stable/2958783
7) Neumann, M.H., (1998),"Strong approximation of density estimators for weakly dependent observation by density estimators from independent observations", Annals of Statistics, V26, no 5, 2014-2048
8) Ruelle, D, (1976), "A measure associated with axiom A attractors", American Journal of Mathematics, V98, no 3, 619-654
9) Bonetto, F., Galavotti, G., Giuliani, A. and Zamponi, F. (2006),"Chaotic Hypothesis, Fluctuation Theorem and Singularities", Journal of Statistical Physics.V123, no 1. DOI: 10.1007/s10955-006-9047-5
10) Galavotti, G. (1996), "Chaotic Hypothesis: Onsager Reciprocity and Fluctuation-Dissipation Theorem", Journal of Statistical Physics, V84, No5/6, 899-925
11) Ye, H., and Sugihara G.,(2016), "Information leverage in interconnected ecosystems, overcoming the curse of dimensionality", Science, Vol 353, issue 6502, 922-925
12) Deyle E.R. Sugihara, G (2011), Generalized Theorems for Nonlinear State Space Reconstruction, PLoS ONE 6(3), e18295, doi:10.1371/journal.poine.0018295
13) Lalley, 1991, "Beneath the noise, chaos" Annals of Statistics 1999, V27, #2, 461-479
14) Judd, 2007, "Failure of maximum likelihood methods for chaotic dynamic systems", Phys. Rev. E, 75 039210
15) Efron B., Hastie, T., Johnstone, I. Tibshirani, R., (2004), "Least Angle Regression", Annals of Statistics 32,(2), 407-499, DOI: 10.1214/009053604000000067
16) Dempster, Laird, and Rubin, "Maximum likelihood from incomplete data with the EM algorithm", (1977), JRSS B, 1-22

Detailed Methods:

Software: I've included the R source code for the analysis presented in this package along with the R data files used.

There are 3 source codes for R (3.2.1 or 2), datamake.pck contains source code for pulling together all the 2 data sets consisting of standardized seasonal data for precipitation and temperature (each weather station), the multivariate enso index, the pacific decadal oscillation, the Indian ocean dipole and the arctic oscillation, and the sunspot data (from sunspot.month in the stats package in R). These, are from 1963 through 2012. Winter is considered December(previous year) January and February. The source code "reproduce.pck" does the main calculation (and is set up to work on windows with a machine with at least 8 virtual processors available) and maketable.pck is used to calculate statistical tests and create the final plots of the program.

Data construction programs

```
dum<-seas.make()

F.nat0full<-dum[,c(12,13,5:8,10)]

H.nat0full<-dum[,c(3:8,10)]

my.scale<-function(vec){(vec-mean(vec))/sd(vec)}

F.nat0full0<-apply(F.nat0full,2,my.scale)

H.nat0full0<-apply(H.nat0full,2,my.scale)
```

#NOTE following depends on the old R ncdf package, not ncdf4, I need to update it for newer data sets

F.nat0full0 and H.nat0full0 are data used in program for those who don't need to reconstruct the original scaled data,

For reconstructed seasurface data use

https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.ncdc:C00927

```
> datamake.pck

 [1] "datamake.pck"

 [2] "table.reduce1"

 [3] "seasonal.index.attractor.make.gulf.ERSST"

 [4] "seasonal.index.attractor.make.IOD.ERSST"

 [5] "trim.mean.1"

 [6] "zone.extract.ersst"

 [7] "Index.zonegulf"

 [8] "Index.zoneIOD1"

 [9] "Index.zoneIOD2"

[10] "season.sep1"

[11] "seas.adj"
```

[12] "pos"

[13] "seas.make"

[14] "HF.nat0full"

The main work is done in the programs in reproduce.pck where the programs turn the standardized data in seasonal anomaly standardized data, and most of the hard work in creating delay maps, building predictions for each delay map, creating residuals and developing the densities are done.

Building the predictive densities programs

#necessary libraries e1071, lars, caret, parallel

This line of code constructs densities for lags from 1 season ahead to 4 for the 9 years being predicted All the other programs are for this analysis are incorporated under "my.uberloop.natsnp.clean"

> **my.natsnpalt.1.4.clean<-my.uberloop.natsnp.clean(1,4)**

> reproduce.pck

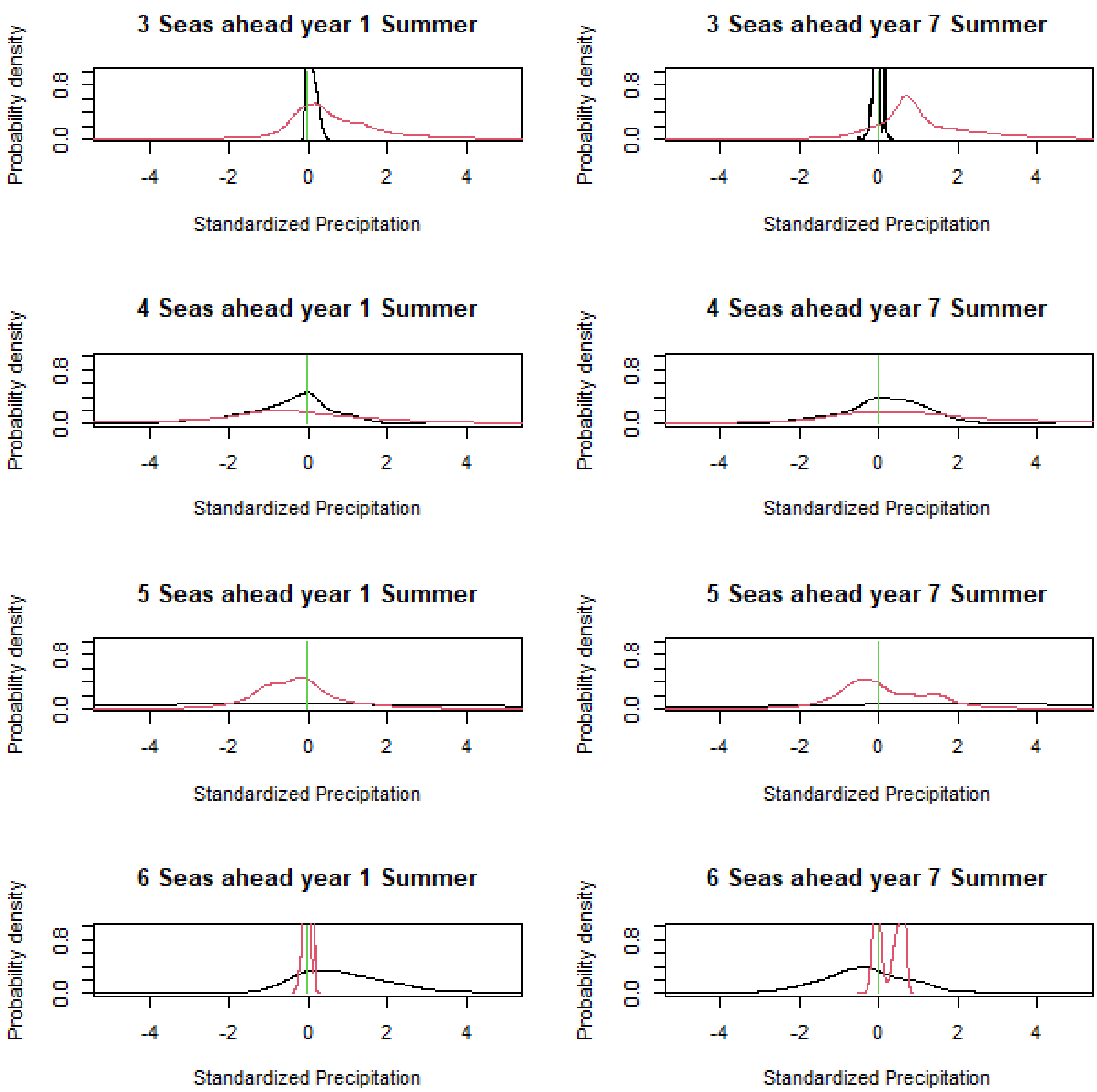
```
 [1] "reproduce.pck"                                  "F.nat0full0"
 [3] "H.nat0full0"                                    "my.uberloop.natsnp.clean"
 [5] "my.ubercompnatsimseas.choice.snp.Temp.clean"
"my.ubercompnatsimseas.choice.snp.Precip.clean"
 [7] "my.overloopHFprecip.snsp.progliknatsimseas.choice.snp.clean"
"my.pair.loop.simH.ext.dens.nonanom.HF.perc.progliknatseas.clean"
 [9] "analog.llik"                                    "my.knormdens"
[11] "my.knorm"
"my.test.chaos.reg4nn.measure.orig.augH.movie.percseas.clean"
[13] "seas.adj.pred"                                  "matlag1"
[15] "matlag0"                                        "dim.est.calc"
[17] "dim.est.loop.calc"                              "dim.est"
[19] "disjoint.delaymap.make1"                        "my.test.chaos.reg4nn.child2"
[21] "rm.na.reg"                                      "my.cp.extract.larsnntry2"
[23] "princompreg"                                    "gen.inv1"
```

This final package produces the various analysis and plots, my.paper.figure1 produces figure 3, and my.paper.figure2 produces figure 4.

Final calculations: Analysis and plotting programs

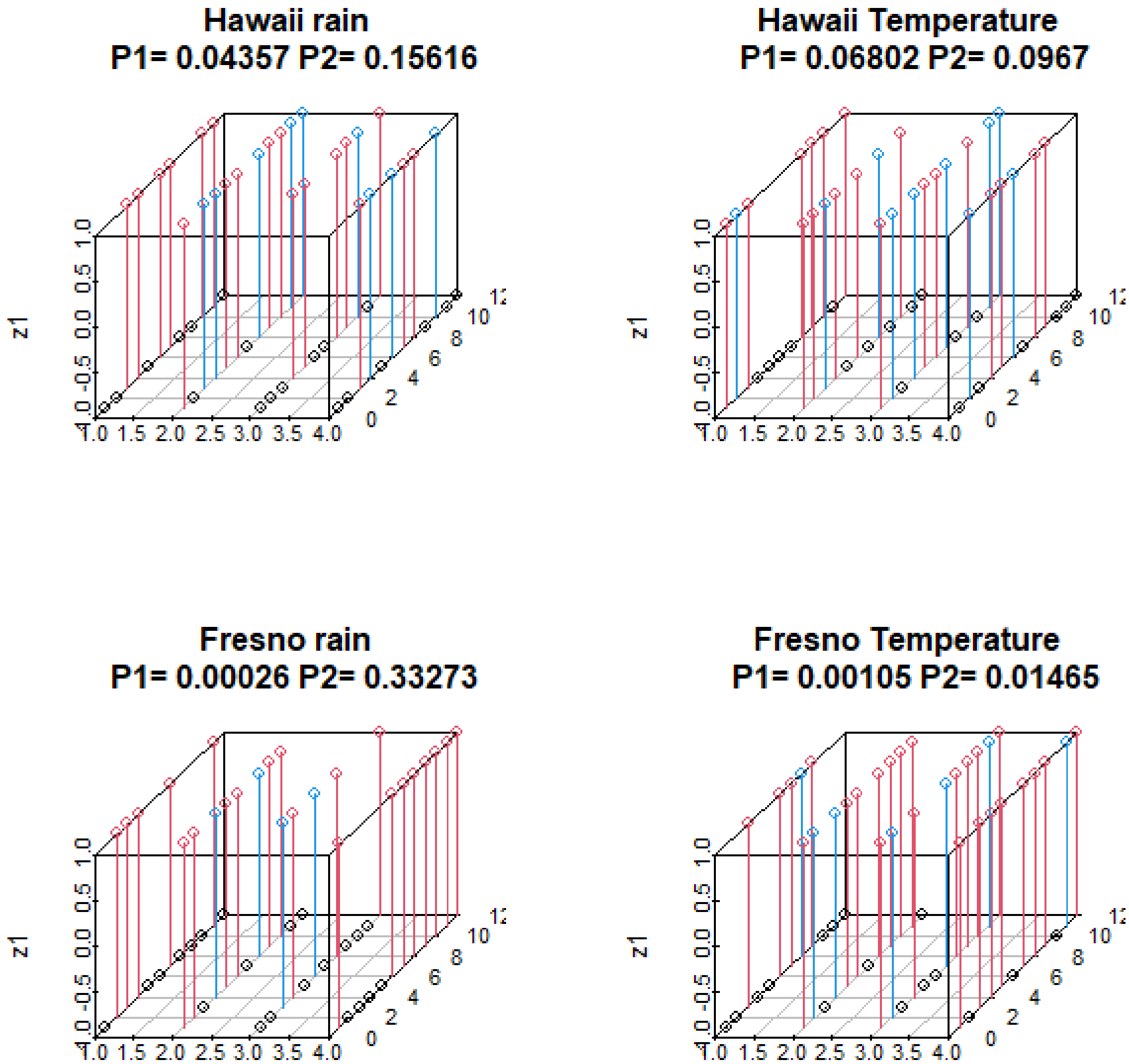
dum1.12.FF<-make.table.ad1.ext(F,F)

```
dum1.12.FT<-make.table.ad1.ext(F,T)
dum1.12.TF<-make.table.ad1.ext(T,F)
dum1.12.TT<-make.table.ad1.ext(T,T)
natsnpfdr.1.12.natlik.t.TT<-make.table.fdr.ext(T,T)
natsnpfdr.1.12.altlik.t.TF<-make.table.fdr.ext(T,F)
natsnpfdr.1.12.natlik.t.FT<-make.table.fdr.ext(F,T)
natsnpfdr.1.12.natlik.t.FF<-make.table.fdr.ext(F,F)
makefig2tot()
```

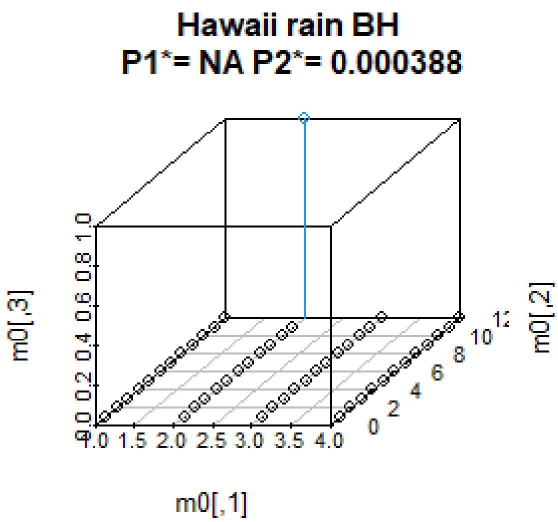


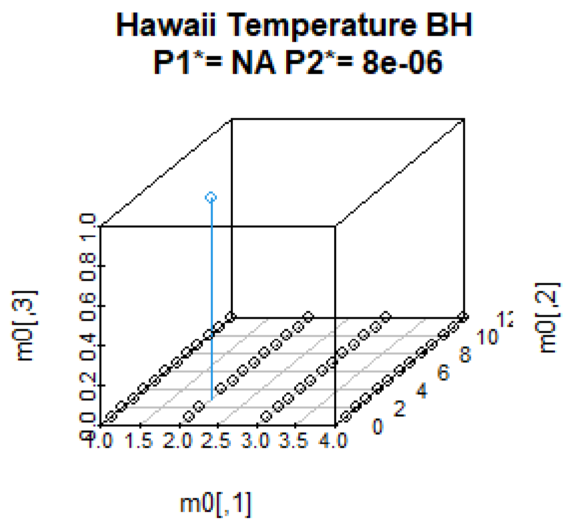
my.paper.figure1()

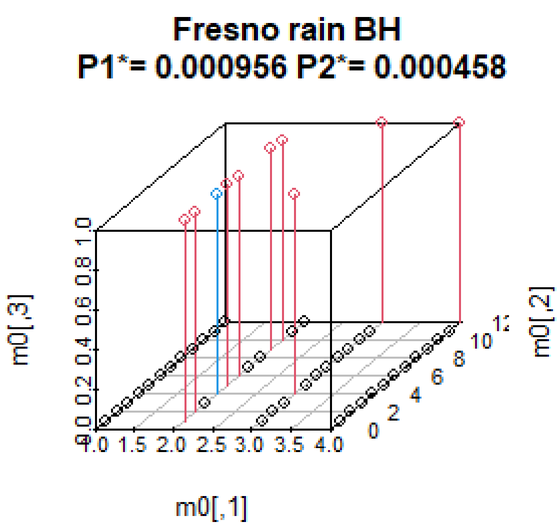
[1] 0.002377619

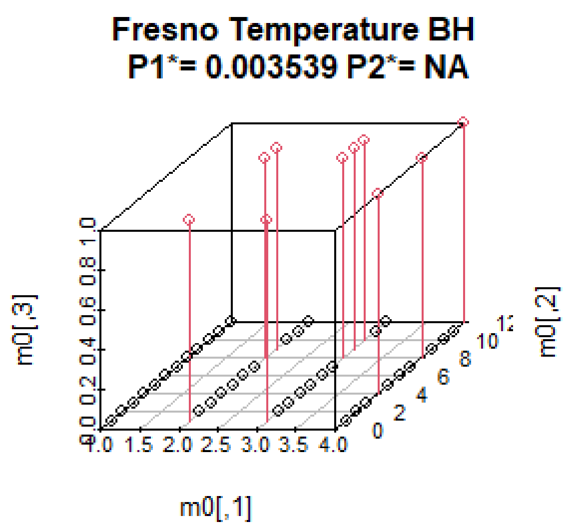


my.paper.figure2(natsnpfdr.1.12.natlik.t.FT,natsnpfdr.1.12.natlik.t.TT,natsnpfdr.1.12.natlik.t.FF,natsnpfdr.1.12.natlik.t.TF,.1)









#Necessary Library scatterplot3d

```
> maketable.pck

[1] "maketable.pck"

[2] "make.table.ad1.ext"

[3] "make.table.fdr.ext"

[4] "format.table.natsnp.byseas.ad1"

[5] "format.table.natsnp.byseas.a.alt"

[6] "lik.mix.ratcorr1e.byseas.ad1"
```

```
 [7] "pinttrans.check"
 [8] "pinttrans"
 [9] "lik.mix.ratcorr1e.byseas.a.alt"
[10] "fdr"
[11] "llik.check.perm"
[12] "my.paper.figure1"
[13] "my.paper.figure2"
[14] "pvalplotpred2"
[15] "pvalplotpred1"
[16] "makefig2tot"
[17] "makefig2"
```