## Wine fermentation by SC and SK species dissected by predictive modeling

Created by David Saque Henriques

Last revised : December 8, 2017

Contacts: davidh@iim.csic.es , ebalsa@iim.csic.es

This distribution is intended to reproduce the computational pipeline applied in the modeling of wine fermentation by *Saccharomyces cerevisiae* and *S. kudriavzevii* at low temperatures.

The implementation is based on AMIGO2 toolbox and exploits the possibility of parallelizing several tasks. For this purpose it is prepared to run in a LINUX cluster by means of MEX files. Its use requires knowledge on LINUX (bash) and Matlab programming. The codes were implemented and tested in Matlab version 2016a.

Alternatively, for the purpose of simulation or parameter estimation, users may solve for candidate models by directly using AMIGO2 toolbox and either C or full MATLAB coding. It should be noted that MATLAB implementation will be computationally more demanding.

### REQUIREMENTS AND INSTALLATION NOTES to run the full modeling pipeline

**LINUX**: The code provided here is designed to run in a LINUX cluster using the Open Grid queue system. Running the whole pipeline in a different OS may require major efforts. Some tasks are automated with the help of shell scripts and a queue system. Cluster configurations vary a lot. However, with moderate effort, the code provided here can be adapted to run in other LINUX machines.

### Configuration

1) Before running the pipeline check your mex system is configured in matlab. You can do this by typing:

mex -setup

2) After configuring the mex options, the mex file needs to be edited. You can find the location of this file by typing:

cc = mex.getCompilerConfigurations

edit(cc.MexOpt)

Remove the -ansi flag (should be found in two different locations) from the mexOpts file.

3) Compile libAMIGO by running: AMIGO\_linux\_macOS\_build.

Hint: in case something fails, check the Makefile under, AMIGOR2017a/Kernel/libAMIGO.

5) Run AMIGO\_Startup

## CODE ORGANIZATION

The relevant scripts and data files relevant to the present work are located under: AMIGOR2017a/Inputs/Yeast

load\_data.m: Loads the data in DATA.csv and model.csv.

compile\_model.m: Compiles the model stored in model.csv

fit\_model.m: fits the model in model.csv. Requires the model has been previously compiled.

Bootstrap.m: If the variable BOOTSTRAP is set to 1, this function creates a sub-sample of the experimental data.

fitModels : Fit all model variants iteratively and store the results.

plotModels : Plot all model variants iteratively.

**MODEL13:** This folder corresponds to model R2 in the main text.

**MODEL18:** This folder corresponds to model R3 in the main text.

**MODEL24:** This folder corresponds to model R1 in the main text.

Inside these folders you will find the following files:

launch\_optimization.sh: Schedule 100 instances of run\_optimization.sh.

launch\_validation.sh: Schedule 100 instances of run\_validation.sh.

run\_optimization.sh: Run one instance of the optimization in matlab.

run\_validation.sh: Run one instance of the cross-validation procedure in matlab.

model.csv: The model.

DATA.csv: The experimental data.

analize\_optimization.m: Analyze the results from the ensemble parameter estimation.

analize\_validation.m: Analyze the results from the ensemble cross-validation.

**MODELS**

The folder MODELS contains all model variants tested during this work.

MODEL1.csv … MODEL24.csv: Model variants.

MODELS.xls: Akaike scores and reduction steps for the relevant models.

## Training a single model

The folder AMIGOR2017a/Inputs/Yeast/MODELS contains a number of models tested during the model reduction phase. To train one of these models:

1) Copy the file MODELX.csv to AMIGOR2017a/Inputs/Yeast/model.csv.

2) Run compile\_model in the Matlab command line.

3) Run the command fit\_model(inputs) .

4) Update #ESTIMATEPARS in model.csv with the parameters obtained.

## Plot Trajectories for a Model

1) Copy the file MODELX.csv to AMIGOR2017a/Inputs/Yeast/model.csv.

2) Run compile\_model in the Matlab command line.

3) Run the command plotCurrentModel

## Ensemble Training and Cross-Validation

MODELX (MODEL 18, 13 and 24) contains the scripts used in the ensemble parameter estimation and cross-validation procedure. Additionally log files and other results are stord in this folder.

1) Copy the folder MODELX to the cluster.

2) Make sure launch\_optimization.sh and run\_optimization.sh have execution permissions and are compatible with your job scheduler. Otherwise modify these appropriately.

2) Open Matlab

3) Go to the AMIGOR2017a directory.

4) Run AMIGO\_Startup

5) Move to AMIGOR2017a/Inputs/Yeast

6) Add the current directory to the path: addpath(pwd)

7) Move to the folder MODELX

8) Run the script compile\_model. If problems arise during compilation, check the CSV is being imported correctly and that you have configured the MEX.

9) Exit Matlab.

10) Run the command launch\_optimization.sh

The procedure to launch the cross-validation is exactly the same. However, in this case, the relevant .sh scripts are called launch\_validation and run\_validation.sh

## Analyzing results from Cross-Validation and Training

1) Make sure all jobs in the cluster have finished.

2) If you analyze the results in your local machine, make sure to compile the model again.

3) Run the script analize\_optimization.m or analize\_validation.m.