Supporting Information

Evaluation of the Open Source Process Simulator DWSIM for Bioprocess Simulation

Siddhi Sreemahadevan[1*], Krishnapriya Velusamy Sivakumar[1], Menjith Palanisamy[1]

[1] Department of Biotechnology, PSG College of Technology, Avinashi Road, Coimbatore, Tamil Nadu 641004, India

[*] Corresponding author, e-mail: sid.bio@psgtech.ac.in

Part A

REACTION AND CONVERSIONS

Table A1 Pretreatment reactions and conversions [18]

Sl No.	REACTION	REACTANT	% CONVERSION
1	$(Glucan)_n$ + n $H_2O \rightarrow$ n Glucose	Glucan	9.9%
2	$(Glucan)_n$ + n $H_2O \rightarrow$ n Glucose Oligomer	Glucan	0.3%
3	Glucan)n $\rightarrow$ n HMF + 2n H_2O	Glucan	0.3%
4	Sucrose $\rightarrow$ HMF + Glucose + 2 H_2O	Sucrose	100%
5	$(Xylan)_n$ + n $H_2O \rightarrow$ n Xylose	Xylan	90.0%
6	$(Xylan)_n$ + m $H_2O \rightarrow$ m Xylose Oligomer	Xylan	2.4%
7	$(Xylan)_n \rightarrow$ n Furfural + 2n H_2O	Xylan	5.0%
8	Acetate $\rightarrow$ Acetic Acid	Acetate	100%
9	$(Lignin)_n \rightarrow$ n Soluble Lignin	Lignin	5.0%

Table A2 Enzymatic Hydrolysis Reactions and conversions [18]

Sl. No.	REACTION	REACTANT	%CONVERSION
1	$(Glucan)_n \rightarrow$ n Glucose Oligomer	Glucan	4.0%
2	$(Glucan)_n$ + ½n $H_2O \rightarrow$ ½n Cellobiose	Glucan	1.2%
3	$(Glucan)_n$ + n $H_2O \rightarrow$ n Glucose	Glucan	90.0%
4	Cellobiose + $H_2O \rightarrow$ 2 Glucose	Cellobiose	100%

Table A3 Co-fermentation reactions and conversions [18]

Sl. No.	REACTION	REACTANT	% CONVERSION
1	Glucose $\rightarrow$ 2 Ethanol + 2 CO_2	Glucose	95.0%
2	Glucose + 0.047 CSLa + 0.018 DAP $\rightarrow$ 6 *Z. mobilis* + 2.4 H_2O	Glucose	2.0%
3	Glucose + 2 $H_2O \rightarrow$ 2 Glycerol + O_2	Glucose	0.4%
4	Glucose + 2 $CO_2 \rightarrow$ 2 Succinic acid + O_2	Glucose	0.6%
5	Xylose $\rightarrow$ 5 Ethanol + 5 CO_2	Xylose	85.0%
6	Xylose + 0.039 CSL + 0.015 DAP $\rightarrow$ 5 *Z. mobilis* + 2 H_2O	Xylose	1.9%
7	2 Xylose + 5 $H_2O \rightarrow$ 5 Glycerol + 2.5 O_2	Xylose	0.3%
8	Xylose + $H_2O \rightarrow$ Xylitol + 0.5 O_2	Xylose	4.6%
9	3Xylose + 5 $CO_2 \rightarrow$ 5 Succinic Acid + 2.5 O_2	Xylose	0.9%

Table A4 Seed train reactions and conversions [18]

Sl. No	REACTION	REACTANT	% CONVERSION
1	Glucose $\rightarrow$ 2 Ethanol + 2 CO_2	Glucose	90.0%
2	Glucose + 0.047 CSLa + 0.018 DAP $\rightarrow$ 6 *Z. mobilis* + 2.4 H_2O	Glucose	4.0%
3	Glucose + 2 $H_2O\rightarrow$ 2 Glycerol + O_2	Glucose	0.4%
4	Glucose + 2 $CO_2\rightarrow$ 2 Succinic acid + O_2	Glucose	0.6%
5	3Xylose $\rightarrow$ 5 Ethanol + 5 CO_2	Xylose	80.0%
6	Xylose + 0.039 CSL + 0.015 DAP $\rightarrow$ 5 *Z. mobilis* + 2 H_2O	Xylose	4.0%
7	3Xylose + 5 $H_2O \rightarrow$ 5 Glycerol + 2.5 O_2	Xylose	0.3%
8	3 Xylose + 5 $CO_2 \rightarrow$ 5 Succinic Acid + 2.5 O_2	Xylose	0.9%
9	Xylose + $H_2O\rightarrow$ Xylitol + 9.5 O_2	Xylose	4.6%

Part B

PYTHON SCRIPTS FOR CUSTOM UNITS

B.1 Basic scripts for getting properties of input stream

```
import math                      #import math library function
from DWSIM import *         #import all DWSIM default functions
from System import Array   #import Array
import System
feed = ims1                                         #ims1 indicates the first input stream ; NOTE: if ims2 then 2nd input
stream then goes on)
T = ims1.Phases[0].Properties.temperature     # Read the Temperature value of input stream
P = ims1.Phases[0].Properties.pressure          # Read the pressure of input stream
n = int(feed.GetNumCompounds())               # Get the number of components in the feed stream
ids = feed.ComponentIds                            # Get compound IDs in the feed stream
inflows = feed.GetProp("flow" ,"Overall", None, "", "mass")   # Get the mole flow and mole fraction for each
component in the feed
Totalflow= feed.GetProp (“totalFlow”,”Overall”,None,””,”mass”)      #Get total flow of stream
```

B.2 Basic scripts for getting properties of output stream

```
for i in range(n):
massfrac[i]=inflows[i]/wf                          #calculate the mass fraction of the newly calculated    flowrates
xwoarr= Array[float](massfrac)                 #converts the mass fraction calculated to floating values as the DWSIM
package supports floating points
xmo = feed.PropertyPackage.AUX_CONVERT_MASS_TO_MOL(xwoarr)
                                                            #converts the mass fraction calculated above to mole fraction
oms1.SetProp("fraction", "Overall", None, "", "mole", xmo)      #set the new mole fraction values to output stream
oms1 indicates output stream 1 .(if oms2 the output stream 2 and goes on..)
oms1.Phases[0].Properties.temperature = System.Nullable[System.Double](T)       #set Temperature of the output
stream
oms1.Phases[0].Properties.pressure = System.Nullable[System.Double](P)             #set Pressure of the output
stream
oms1.Phases[0].Properties.massflow = System.Nullable[System.Double](wf)          #set total flow of the output
stream
```

B.3 Python script for pretreatment tank

```
xylaan=inflows[4]
glucoose=inflows[7]
other_sugar=inflows[13]
cellulose=inflows[8]
#sugar oligomers
inflows[14]=(0.003*cellulose)+(0.024*xylaan)
#cellulose:
cellulose_reacted=inflows[8]*0.099
water_required=cellulose_reacted*0.9
if inflows[1]>water_required:
    inflows[8]=inflows[8]*0.901
    inflows[1]=inflows[1]-water_required
    inflows[7]=water_required+inflows[13]
else:
    glucose_produced=inflows[1]/.9
    inflows[8]=inflows[8]-(0.994255*glucose_produced)
    inflows[7]=glucose_produced+inflows[13]
    inflows[1]=0
#other sugars
oligos=inflows[17]/3
inflows[13]=inflows[13]+1.8*(0.003*cellulose)+(oligos*0.90*1.11*2)+(oligos*0.90*1.136)
#xylose:
xylan_reacted=inflows[4]*0.90
water_required=xylan_reacted*0.8
if inflows[1]>water_required:
    inflows[4]=inflows[4]*0.026
    inflows[3]=water_required
    inflows[1]=inflows[1]-water_required
else:
    xylose_produced=inflows[1]/0.8
    inflows[4]=inflows[4]-(0.8799*xylose_produced)
    inflows[3]=xylose_produced
    inflows[1]=0
#lignin
inflows[6]=0.95*inflows[6]
#furfural
init=inflows[11]
inflows[11]= ((xylaan*0.05)*1.05904)+inflows[11]   #furfurals
#water
if inflows[1]!=0:
    inflows[1]=inflows[1]-(xylaan*0.025)-(glucoose*0.05)+((inflows[11]-
init)*2)+(other_sugar*2)+((35.37*(0.003*cellulose))*2)
else:
    inflows[1]=((inflows[11]-init)*2)+(other_sugar*2)+((35.37*(0.003*cellulose))*2)
massfrac=[0]*n
wf=sum(inflows)
```

B.4 Python script for saccharification and fermentation tank

```
feed = ims1
seed=ims2
inflows1 = feed.GetProp("flow" ,"Overall", None, "", "mass")                              # mol/s
inflows2=seed.GetProp("flow" ,"Overall", None, "", "mass")
vent=[0]*n
sachrri=[0]*n
ferment=[0]*n
cellulose=inflows1[8]
water=inflows1[1]+inflows2[1]
inflows1[1]=inflows1[1]+inflows2[1]
#cellulose updation
sachrri[8]=inflows2[8]
vent[8]=0
cellulose_reacted=cellulose*0.90
water_required=cellulose_reacted*0.9
if water_required<inflows1[1]:
    inflows1[7]=inflows1[7]+water_required
   # inflows1[1]=inflows1[1]-water_required
    inflows1[8]=inflows1[8]-cellulose_reacted
    cellulose_reacted=cellulose*0.012
    water_required=cellulose_reacted*0.47
```

```
        if water_required<inflows1[1]:
            inflows1[1]=inflows1[1]-water_required
            inflows1[8]=inflows1[8]-cellulose_reacted
            cellobiose=water_required
            water_required=cellobiose
            if water_required<inflows1[1]:
                inflows1[7]=inflows1[7]+(2*cellobiase)
                inflows1[1]=inflows1[1]-water_required
            else:
                cellobiose_reacted=inflows1[1]
                inflows1[13]=inflows1[13]+(cellobiose-cellobiose_reacted)
                inflows1[1]=0
                inflows1[7]=inflows1[7]+(cellobiose_reacted*2)
        else:
            cellulose_reacted=inflows1[1]/0.5
            inflows1[1]=0
            inflows1[8]=inflows1[8]-cellulose_reacted
            inflows1[13]=inflows1[13]+(cellulose_reacted*2)
    else:
        cellulose_reacted=inflows1[1]/0.9
        inflows1[1]=0
        inflows1[7]=inflows1[7]+(cellulose_reacted*0.9)
        inflows1[8]=inflows1[8]-cellulose_reacted
    inflows1[7]=inflows1[7]+inflows2[7]
    ferment[8]=inflows1[8]
    #sugar oligomers
    sachrri[14]=inflows2[14]
    ferment[14]=inflows1[14]+(cellulose*0.04)
    vent[14]=0
    #other sugars
    sachrri[13]=inflows2[14]
    vent[13]=0
    other_sugars_to_xylose=inflows1[13]*0.409
    ferment[13]=inflows1[13]-other_sugars_to_xylose
    #xylose
    xylose_input=inflows1[3]+inflows2[3]+other_sugars_to_xylose
    unreacted_xylose=0.15*xylose_input
    vent[3]=0
    sachrri[3]=unreacted_xylose*0.608
    ferment[3]=unreacted_xylose*0.391
    eth_xylose=(0.85*xylose_input)*1.6
    #glucose
    unreacted_glucose=total_glucose*0.05
    ferment[7]=unreacted_glucose*0.149
    vent[7]=0
    sachrri[7]=unreacted_glucose-ferment[7]
    #water updation
    total_water=inflows1[1]+(2.4*0.04*total_glucose)+(2*0.04*xylose_input)
    final_water=total_water-(total_glucose*0.004*2)-(xylose_input*5*0.004)
    vent[1]=final_water*0.00095
    sachrri[1]= final_water*0.0907
    ferment[1]=final_water*0.9082
    #ethanol
    eth_from_ferment=(final_xylose*0.85*1.6)+(final_glucose*0.95*2)+inflows1[2]+inflows2[2]
    sachrri[2]=eth_from_ferment*0.00001
    vent[2]=0.015*eth_from_ferment
```

```
ferment[2]=eth_from_ferment*0.9848
pt=sum(inflows2)
#lignin
vent[6]=0
ferment[6]=inflows1[6]
sachrri[6]=inflows2[6]
#carbondi oxide
total_co2=eth_from_ferment-inflows1[2]-inflows2[2]
sachrri[0]=0
ferment[0]=total_co2*0.027
vent[0]=total_co2*0.972
#protein
total_protein=inflows1[5]+inflows2[5]
sachrri[5]=total_protein*0.0943
ferment[5]=total_protein*0.9056
vent[5]=0
#cellmass
total_biomass=(xylose_input*0.04*5)+(0.04*inflows1[7]*6)
ferment[9]=total_biomass*0.198
sachrri[9]=0.00231*total_biomass
vent[9]=0
#furfurals
sachrri[11]=inflows2[11]
ferment[11]=inflows1[11]
#xylan
ferment[4]=inflows1[4]
sachrri[4]=inflows2[4]
vent[4]=0
sumfer=sum(ferment)
sumvent=sum(vent)
sumsach=sum(sachrri)
massfrac1=[0]*n
massfrac2=[0]*n
massfrac3=[0]*n
for i in range(n):
    massfrac1[i]=ferment[i]/sumfer
    massfrac2[i]=vent[i]/sumvent
    massfrac3[i]=sachrri[i]/sumsach
# convert IronPython lists to .NET arrays
xwoarr1 = Array[float](massfrac1)
xwoarr2 = Array[float](massfrac2)
xwoarr3 = Array[float](massfrac3)
# Use Property Package's mass-to-mole fraction conversion function because
# You'll have to set the mole (not mass) fractions in the output streams...
xmo1 = feed.PropertyPackage.AUX_CONVERT_MASS_TO_MOL(xwoarr1)
xmo2 = feed.PropertyPackage.AUX_CONVERT_MASS_TO_MOL(xwoarr2)
xmo3 = feed.PropertyPackage.AUX_CONVERT_MASS_TO_MOL(xwoarr3)
# Create vectors
oms1.SetProp("fraction", "Overall", None, "", "mole", xmo1)
oms1.Phases[0].Properties.temperature = System.Nullable[System.Double](T)
oms1.Phases[0].Properties.pressure = System.Nullable[System.Double](P)
oms1.Phases[0].Properties.massflow = System.Nullable[System.Double](sumfer)
oms2.SetProp("fraction", "Overall", None, "", "mole", xmo2)
oms2.Phases[0].Properties.temperature = System.Nullable[System.Double](T)
oms2.Phases[0].Properties.pressure = System.Nullable[System.Double](P)
oms2.Phases[0].Properties.massflow = System.Nullable[System.Double](sumvent)
```

```
oms3.SetProp("fraction", "Overall", None, "", "mole", xmo3)
oms3.Phases[0].Properties.temperature = System.Nullable[System.Double](32+274)
oms3.Phases[0].Properties.pressure = System.Nullable[System.Double](P)
oms3.Phases[0].Properties.massflow = System.Nullable[System.Double](sumsach)
```

B.5 Python script for seed ferment0r

```
feed = ims1
seed=ims2
deed=ims3
# Get temperature and pressure of the feed
# Notice that the values returned are single element vectors, not scalars
T = ims1.Phases[0].Properties.temperature
P = ims1.Phases[0].Properties.pressure
# Get the number of components in the feed stream
n = int(feed.GetNumCompounds())
# Get compound IDs in the feed stream
ids = feed.ComponentIds
# Get the mole flow and mole fraction for each component in the feed
inflows1 = feed.GetProp("flow" ,"Overall", None, "", "mass")                              # mol/s
inflows2 = seed.GetProp("flow" ,"Overall", None, "", "mass")
inflows3=  deed.GetProp("flow" ,"Overall", None, "", "mass")
massfrac1=[0]*n
massfrac2=[0]*n
total_water=inflows1[1]+inflows2[1]
total_protein=inflows1[5]+inflows2[5]
inflows1[1]=total_water*0.99980
inflows2[2]=total_water*0.00107
inflows1[5]=total_protein
#biomass
inflows1[9]=(inflows1[7]*0.04*6)+(inflows1[4]*0.04*5)
inflows1[9]=inflows1[9]*0.23
#ethanol
total_eth=(inflows1[7]*0.90*2+inflows1[3]*0.80*1.6)
inflows1[2]=total_eth*0.9814
inflows2[2]=total_eth*0.01856
inflows1[7]=inflows1[7]*0.056
inflows1[3]=inflows1[3]*0.11
```

B.6 Python script for distillation

```
import math
from DWSIM import * # DWSIM namespace is imported automatically by the script tool
from System import Array
import System
feed = ims1
# Get temperature and pressure of the feed
# Notice that the values returned are single element vectors, not scalars
T = ims1.Phases[0].Properties.temperature
P = ims1.Phases[0].Properties.pressure
# Get the number of components in the feed stream
n = int(feed.GetNumCompounds())
# Get compound IDs in the feed stream
ids = feed.ComponentIds
# Get the mole flow and mole fraction for each component in the feed
```

```
inflows = feed.GetProp("flow" ,"Overall", None, "", "mass")                    # mol/s
massfrac1=[0]*n
massfrac2=[0]*n
massfrac3=[0]*n
distillate=[0]*n
vent=[0]*n
distillate[2]=inflows[2]*0.98860
distillate[1]=inflows[1]*0.097
distillate[11]=inflows[11]*0.243
distillate[0]=inflows[0]*0.04918
vent[2]=inflows[2]*0.003
vent[1]=inflows[1]*0.0000607
vent[0]=inflows[0]*0.95
inflows[2]=inflows[2]*0.008386
inflows[1]=inflows[1]*0.89
inflows[11]=inflows[11]*0.745
inflows[0]=0
wf1=sum(distillate)
wf2=sum(inflows)
wf3=sum(vent)
for i in range(n):
    massfrac1[i]=distillate[i]/wf1
    massfrac2[i]=inflows[i]/wf2
    massfrac3[i]=vent[i]/wf3
xwoarr1= Array[float](massfrac1)
xwoarr2=Array[float](massfrac2)
xwoarr3=Array[float](massfrac3)
xmo1 = feed.PropertyPackage.AUX_CONVERT_MASS_TO_MOL(xwoarr1)
xmo2 = feed.PropertyPackage.AUX_CONVERT_MASS_TO_MOL(xwoarr2)
xmo3 = feed.PropertyPackage.AUX_CONVERT_MASS_TO_MOL(xwoarr3)
# Create vectors
oms1.SetProp("fraction", "Overall", None, "", "mole", xmo1)
oms1.Phases[0].Properties.temperature = System.Nullable[System.Double](T+13)
oms1.Phases[0].Properties.pressure = System.Nullable[System.Double](P)
oms1.Phases[0].Properties.massflow = System.Nullable[System.Double](wf1)
oms2.SetProp("fraction", "Overall", None, "", "mole", xmo2)
oms2.Phases[0].Properties.temperature = System.Nullable[System.Double](T)
oms2.Phases[0].Properties.pressure = System.Nullable[System.Double](P)
oms2.Phases[0].Properties.massflow = System.Nullable[System.Double](wf2)
oms3.SetProp("fraction", "Overall", None, "", "mole", xmo3)
oms3.Phases[0].Properties.temperature = System.Nullable[System.Double](T-44)
oms3.Phases[0].Properties.pressure = System.Nullable[System.Double](P)
oms3.Phases[0].Properties.massflow = System.Nullable[System.Double](wf3)
```

Part C

FLOWSHEET OF EACH SECTION

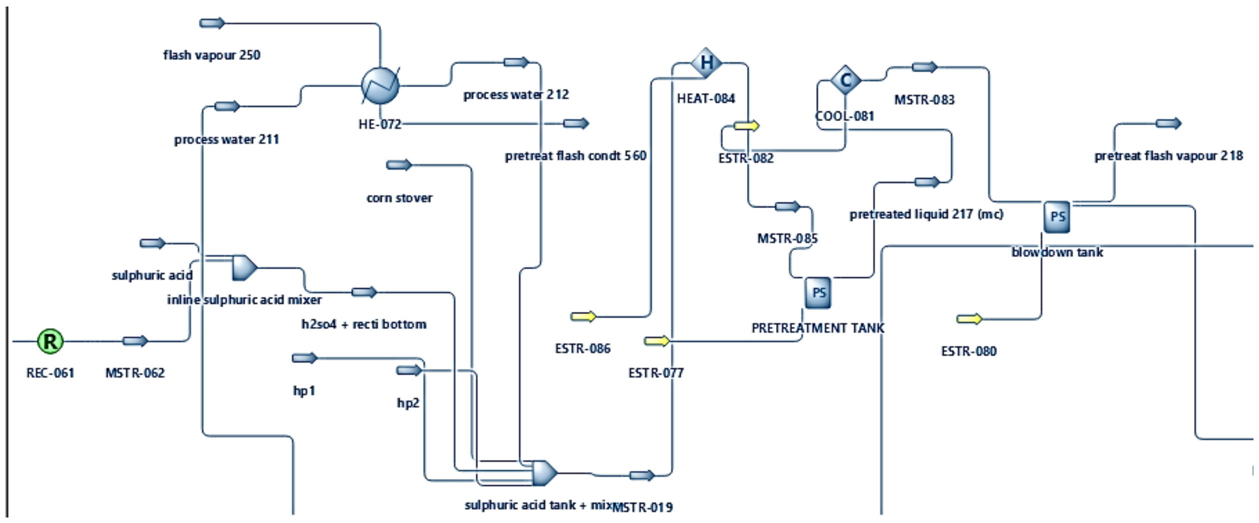

Fig. C1 Flowsheet of pretreatment process in DWSIM

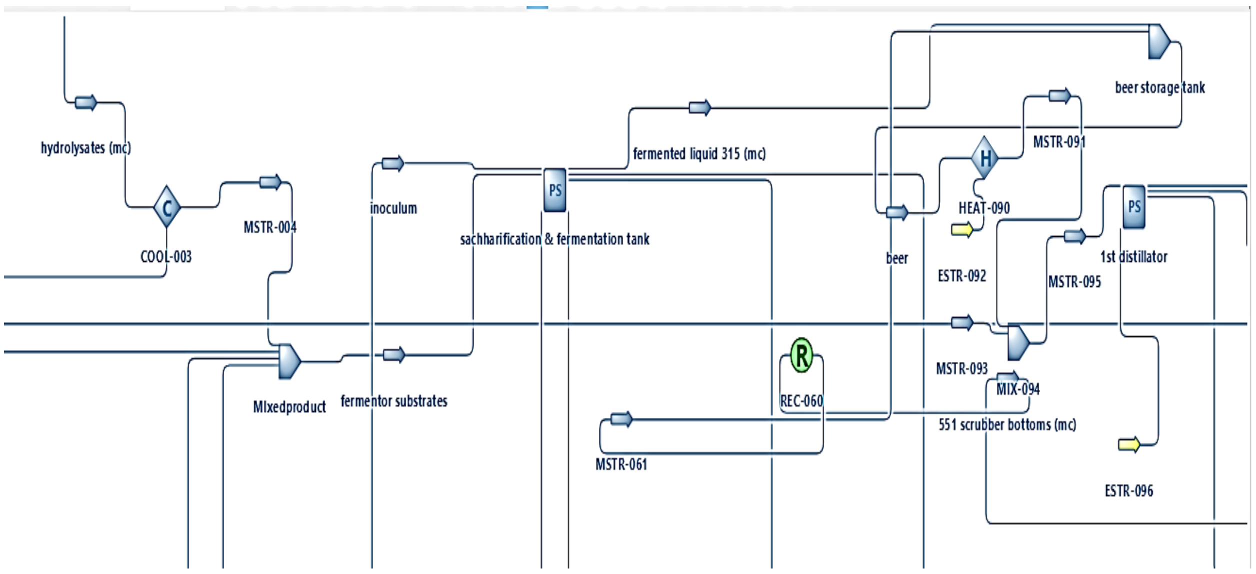

Fig. C2 Flowsheet of saccharification and fermentation in DWSIM

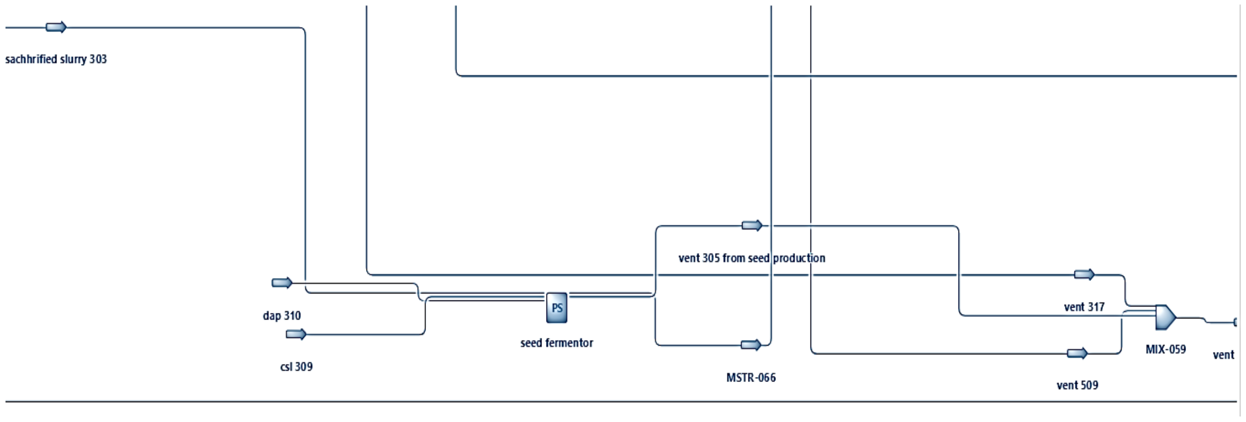

Fig. C3 Flowsheet of seed fermentor in DWSIM

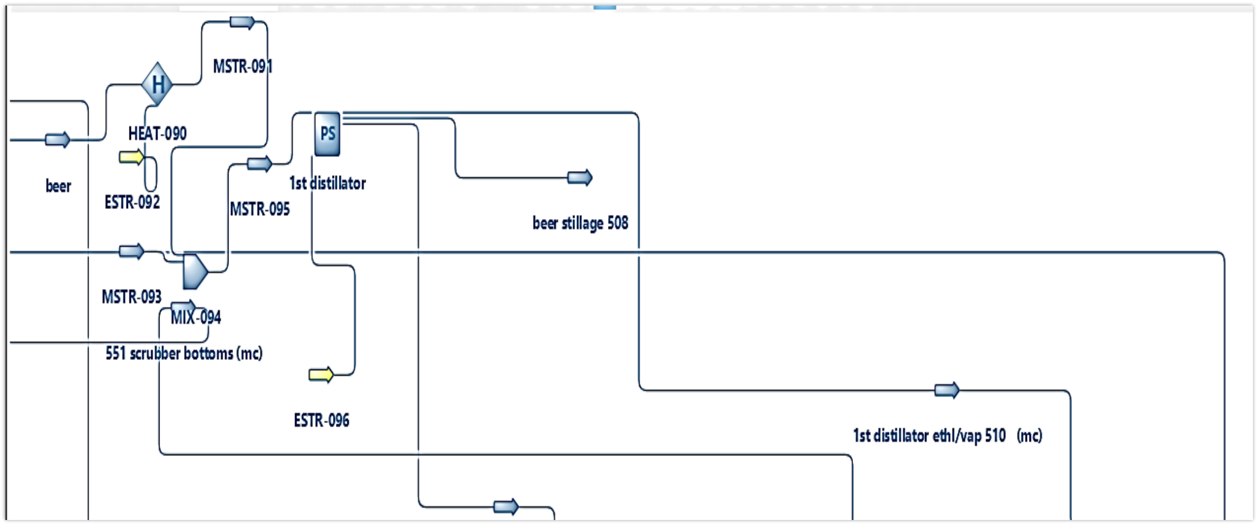

Fig. C4 Flowsheet of the distillation process in DWSIM

Part D

MAIN PROPERTIES OF COMPONENTS ADDED NEW TO DWSIM

D.1 Glucose

BASIC PROPERTIES	**VALUES**	**UNITS**
CAS ID	50-99-7	
Molecular Weight	180	
Critical Temperature	1034	K
Critical Pressure	6631370	Pa
Gibbs Energy of Formation (Ideal Gas at 298.15 K)	-4394	kJ/kg
Enthalpy of Formation (Ideal Gas at 298.15 K)	-5744	kJ/kg
Normal Boiling Point	844	K
Temperature of Fusion (Tf)	423	K
Enthalpy of Fusion @ Tf	105	kJ/mol

D.2 Xylose

BASIC PROPERTIES	**VALUES**	**UNITS**
CAS ID	58-86-6	
Molecular Weight	150.13	
Critical Temperature	1034.02	K
Critical Pressure	6631370	Pa
Gibbs Energy of Formation (Ideal Gas at 298.15 K)	-4395.52	kJ/kg
Enthalpy of Formation (Ideal Gas at 298.15 K)	-5744.89	kJ/kg
Normal Boiling Point	855.48	K
Temperature of Fusion (Tf)	423	K
Enthalpy of Fusion @ Tf	105.19	kJ/mol

D.3. Xylan

BASIC PROPERTIES	VALUES	UNITS
CAS ID	9014-63-5	
Molecular Weight	132.117	
Critical Temperature	1034	K
Critical Pressure	6631370	Pa
Gibbs Energy of Formation (Ideal Gas at 298.15 K)	-4394	kJ/kg
Enthalpy of Formation (Ideal Gas at 298.15 K)	-5744	kJ/kg
Normal Boiling Point	844	K
Temperature of Fusion (Tf)	423	K
Enthalpy of Fusion @ Tf	105	kJ/mol

D.4 Lignin

BASIC PROPERTIES	VALUES	UNITS
CAS ID	9005-53-2	
Molecular Weight	152.149	
Critical Temperature	1034	K
Critical Pressure	6631370	Pa
Gibbs Energy of Formation (Ideal Gas at 298.15 K)	-4394	kJ/kg
Enthalpy of Formation (Ideal Gas at 298.15 K)	-5744	kJ/kg
Normal Boiling Point	844	K
Temperature of Fusion (Tf)	423	K
Enthalpy of Fusion @ Tf	105	kJ/mol

D.5 Other sugars

BASIC PROPERTIES	VALUES	UNITS
CAS ID	7664-93-9	
Molecular Weight	342	
Critical Temperature	1034	K
Critical Pressure	6631370	Pa
Gibbs Energy of Formation (Ideal Gas at 298.15 K)	-4394	kJ/kg
Enthalpy of Formation (Ideal Gas at 298.15 K)	-5744	kJ/kg
Normal Boiling Point	0	K
Temperature of Fusion (Tf)	423	K
Enthalpy of Fusion @ Tf	105	kJ/mol

D.6 Sugar oligomers

BASIC PROPERTIES	VALUES	UNITS
Molecular Weight	180	
Critical Temperature	1034	K
Critical Pressure	6631370	Pa
Gibbs Energy of Formation (Ideal Gas at 298.15 K)	-4394	kJ/kg
Enthalpy of Formation (Ideal Gas at 298.15 K)	-5744	kJ/kg
Normal Boiling Point	844	K
Temperature of Fusion (Tf)	423	K
Enthalpy of Fusion @ Tf	105	kJ/mol

D.7 Protein

BASIC PROPERTIES	VALUES	UNITS
CAS ID	100209-41-4	
Molecular Weight	22.8396	
Critical Temperature	1034	K
Critical Pressure	6631370	Pa
Gibbs Energy of Formation (Ideal Gas at 298.15 K)	-4394	kJ/kg
Enthalpy of Formation (Ideal Gas at 298.15 K)	-5744	kJ/kg
Normal Boiling Point	844	K
Temperature of Fusion (Tf)	423	K
Enthalpy of Fusion @ Tf	105	kJ/mol

D.8 Biomass

BASIC PROPERTIES	VALUES	UNITS
Molecular Weight	23.238	
Critical Temperature	1034	K
Critical Pressure	6631370	Pa
Gibbs Energy of Formation (Ideal Gas at 298.15 K)	-4394	kJ/kg
Enthalpy of Formation (Ideal Gas at 298.15 K)	-5744	kJ/kg
Normal Boiling Point	844	K
Temperature of Fusion	423	K
Enthalpy of Fusion @ Tf	105	kJ/mol