SUPPORTING INFORMATION

Franco Mangone [a], Jimena Ferreira [a,b], Ana I. Torres[a]**

[a] *Instituto de Ingeniería Química, Universidad de la República, Montevideo, 11300, Uruguay*
[b] *Instituto de Computación, Universidad de la República, Montevideo, 11300, Uruguay*

** Corresponding author. *Email address:*aitorres@fing.edu.uy

S1. Screen Captures

S1.1. Aspen Plus Collaborative Environment

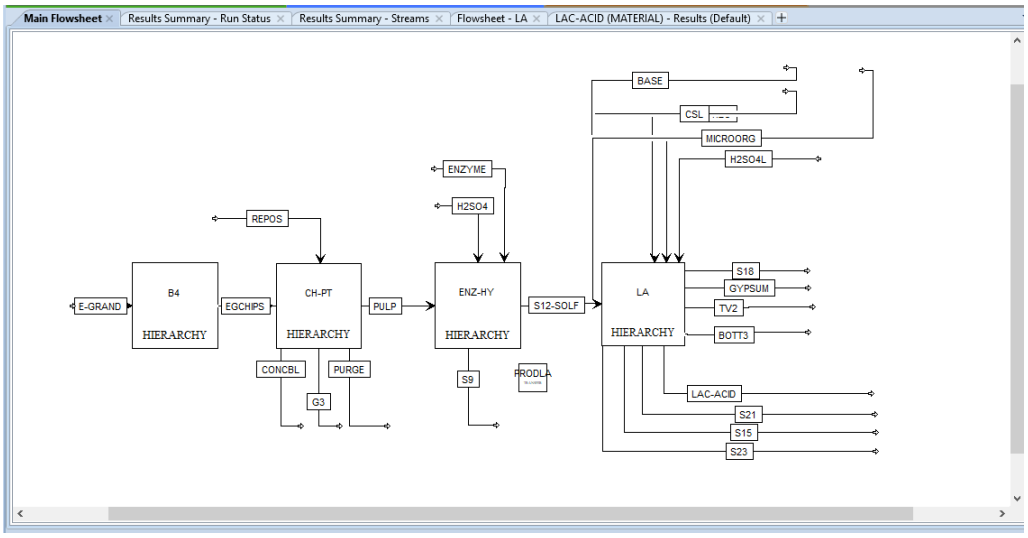

Figure S1. Implementation of the *Eucalyptus Grandis* to Lactic Acid via organosolv pretreatment in the superstructure: Main Flowsheet

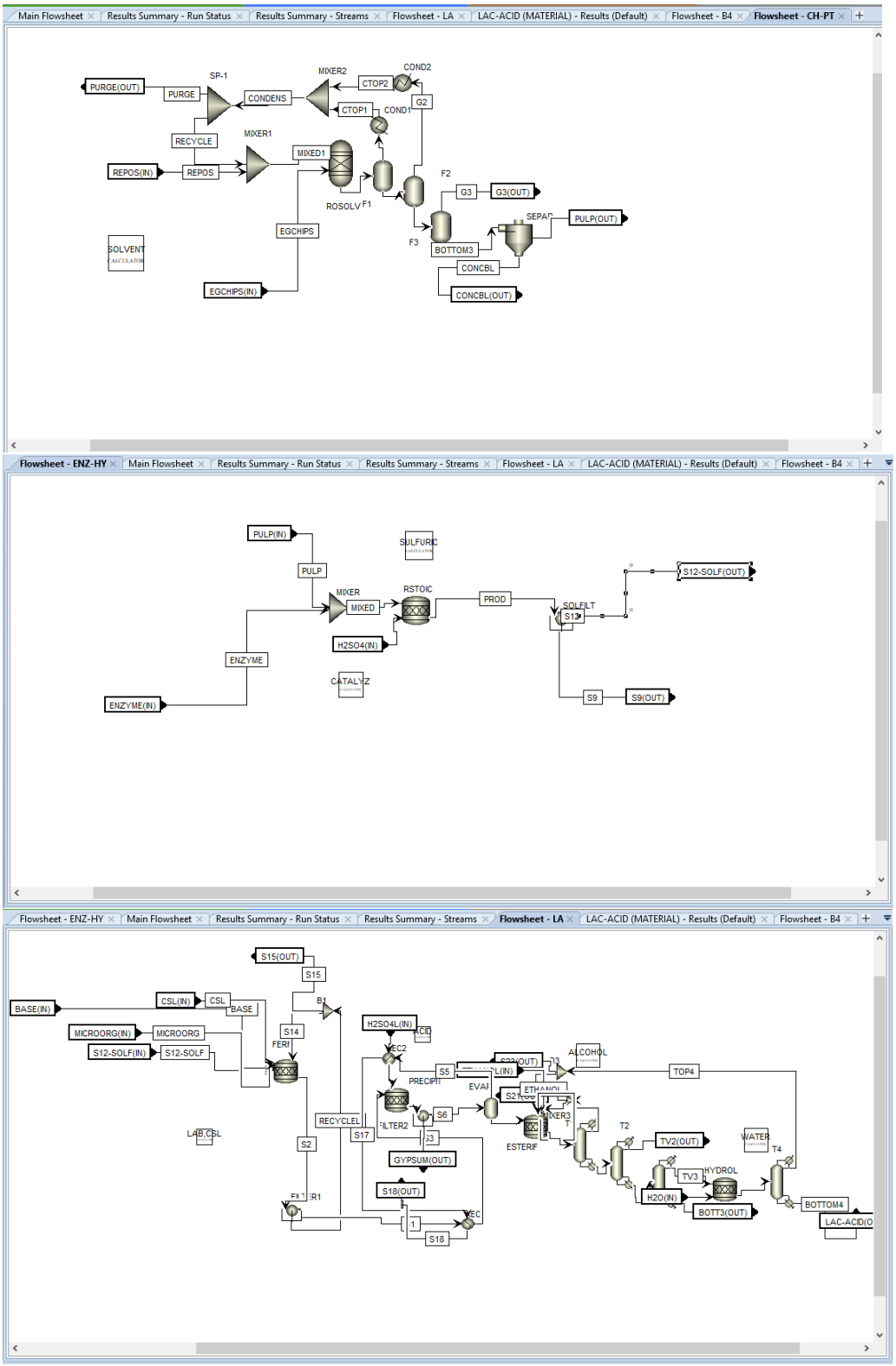

Figure S2. Implementation of the *Eucalyptus Grandis* to Lactic Acid via organosolv pretreatment in the superstructure: Flowsheets for Organosolv pretreatment (Stage 2), Enzymatic Hydrolysis (Stage 3), and Lactic Acid production and purification (Stage 4).

S1.2. Example of data input in BISSO technology block

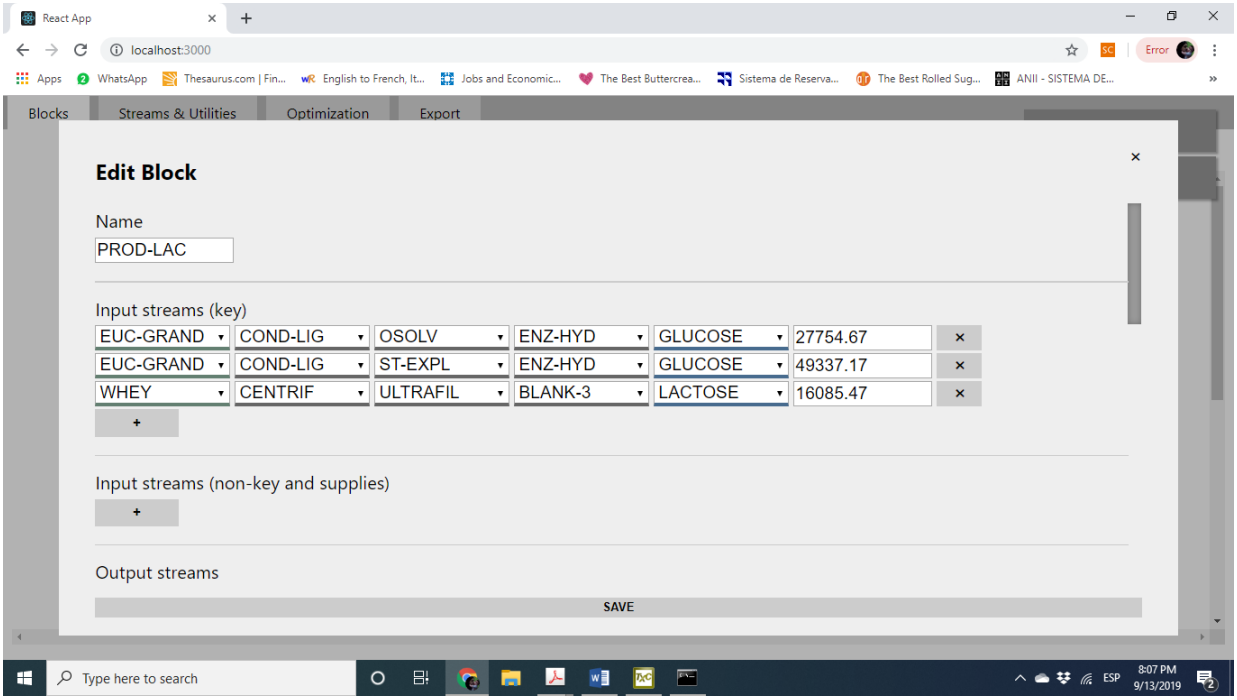

Figure S3. Screen capture showing previous processing pathways that are allowed before fermentation to Lactic Acid.

S2. Example of assignment of order to processing pathways

The example assumes the processing network in Fig.:S4, it consists of two feedstocks, two stage 1 technology blocks and three stage 2 technology blocks.

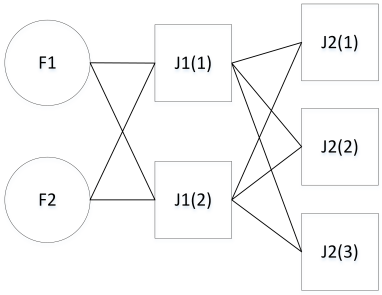

Figure S4. Network used in the assignment example

Hence, there are twelve possible processing pathways and from Eq. 3 variable is computed as:

$$\xi = ord(J2) + card(J2)(ord(J1) - 1) + card(J2)card(J1)(ord(F) - 1) \tag{S1}$$

The assignment order for each pathway is:

Pathway	ord(F)	ord(J1)	ord(J2)	ξ
F1 $\rightarrow$ J11 $\rightarrow$ J21	1	1	1	1
F1 $\rightarrow$ J11 $\rightarrow$ J22	1	1	2	2
F1 $\rightarrow$ J11 $\rightarrow$ J23	1	1	3	3
F1 $\rightarrow$ J12 $\rightarrow$ J21	1	2	1	4
F1 $\rightarrow$ J12 $\rightarrow$ J22	1	2	2	5
F1 $\rightarrow$ J12 $\rightarrow$ J23	1	2	3	6
F2 $\rightarrow$ J11 $\rightarrow$ J21	2	1	1	7
F2 $\rightarrow$ J11 $\rightarrow$ J22	2	1	2	8
F2 $\rightarrow$ J11 $\rightarrow$ J23	2	1	3	9
F2 $\rightarrow$ J12 $\rightarrow$ J21	2	2	1	10
F2 $\rightarrow$ J12 $\rightarrow$ J22	2	2	2	11
F2 $\rightarrow$ J12 $\rightarrow$ J23	2	2	3	12

S3. Including thermochemical biomass conversion processes in BISSO

BISSO is a general tool for generating superstructures for biomass conversion processes. Although, due to our expertise, the examples that we have selected as case studies focus on biochemical transformation, thermochemical bio-refineries are not excluded. In here we provide an example on how the same structure that we have used in Fig. 5, can be used to fit thermochemical biomass conversion processes:

- Stage 1: Conditioning: Any biomass conditioning process such as drying or chipping that is required before the gasification / pyrolysis step should be included here
- Stage 2: Includes all biomass deconstruction processes. Fig. 5 shows pretreatment, if expanded to thermochemical biorefineries, processing steps containing for example gasification, pyrolysis, etc., should be included in here. Compounds after these stage 2 technologies could then be syngas or pyrolysis oil.
- Stage 3: Includes technologies for depolymerization of the compounds after stage 2. These may not be needed in thermochemical processes. In these cases, technologies after Stage 2 will be connected to the Stage 3-Blank technology
- Stage 4 and after: will include all technologies for upgrading intermediates from the thermochemical gasification processes. For example, a Fischer-Tropsch based process to convert syngas to hydrocarbons should be include here.

S4. Files to connect tool with Aspen via Python

S4.1. .json file generated to request data form Aspen


```
 "superstructure.bkp":
 {"feedstock":0,"path":[0],
  "key_stream":[{"aspen_name":"EGCHIPS","problem_name":"CHIPS","value":null}],
"outputs":[{"aspen_name":"PULP","problem_name":"Cellulos","value":null},
 {"aspen_name":"CONCBL","problem_name":"Lignin","value":null},
 {"aspen_name":"G3","problem_name":"Misc","value":null}],
  "supplies":[{"aspen_name":"REPOS","problem_name":"REP-ETHANOL","value":null}]}}
```

S4.2. Python code that loads data/ runs Aspen Plus

```
Created on Mon Mar 25 19:35:15 2019

  @author: Jimena_IIQ_INCO_UdelaR
def run(archive):
        import json
        with open(archive) as json_file:
            data = json.load(json_file)

        import os
        import win32com.client as win32
        aspen = win32.Dispatch('Apwn.Document')

        for name_file in data.keys():
            aspen.InitFromArchive2(os.path.abspath(name_file))
            dict1= data[name_file]
            for var_type in dict1.keys():
                if var_type=='feedstock':
                    None
                elif var_type=='path':
                    None
                elif var_type == 'inputs':
                    lista_in= dict1[var_type]
                    for i in range(len(lista_in)):

                        if lista_in[i]['type']=='MASS-FLOW':

                            legend= lista_in[i]['aspen_name']
                            tipo= lista_in[i]['stream_type']
                            aspen.Tree.FindNode('\Data\Streams\\'
 + legend + '\Input\TOTFLOW\\'+tipo).Value = float(lista_in[i]['value'])
                        elif lista_in[i]['type']=='REYIELD':
                            hierarchy_name=lista_in[i]['in_hierarchy']
                            legend= lista_in[i]['aspen_name']
                            nombre_comp= lista_in[i]['component_name']
                            aspen.Tree.FindNode('\Data\Blocks\\'+hierarchy_name+
  '\\Data\Blocks\\' + legend + '\Input\MOLE_YIEDL\\'+
nombre_comp).Value =float( lista_in[i]['value'])
                        elif lista_in[i]['type']=='RECYCLE':
                            hierarchy_name=lista_in[i]['in_hierarchy']
                            legend= lista_in[i]['aspen_name']
                            reciclo= lista_in[i]['recycle_stream']
                            aspen.Tree.FindNode('\Data\Blocks\\'+hierarchy_name
   +'\\Data\Blocks\\' + legend + '\\Input\FRAC\\'
  + reciclo).Value = float(lista_in[i]['value'])
                        aspen.Engine.Run2()
                elif var_type == 'outputs' or var_type == 'key_stream':

                    lista_out= dict1[var_type]
                    for i in range(len(lista_out)):
```

```
                    legend= lista_out[i]['aspen_name']
                   lista_out[i]['value'] = aspen.Tree.FindNode('\Data\Streams\\'
+ legend + '\Output\MASSFLMX\$TOTAL').Value
            elif var_type== 'supplies':
                lista_out= dict1[var_type]
                for i in range(len(lista_out)):
                    legend= lista_out[i]['aspen_name']
                    lista_out[i]['value'] = aspen.Tree.FindNode('\Data\Streams\\'
+ legend + '\Output\MASSFLMX\$TOTAL').Value

        aspen.close()

    with open('value_'+archive, 'w') as outfile:
        json.dump(data, outfile)

    return None

run('REQUEST.json')
```

S5. Data used run base case simulations

S5.1. In the format used in the GAMS file

```
* Offsite facility parameters ----------
A_off(U) "Offsite facility cost pre-exponential factor" /
          WATER                    9.304
          COOLING-WATER            666666
          REFRIGERANT              6666666
          VAPOR                    1764.3
          ENERGY                   666666
/
b_off(U) "Offsite facility cost exponential factor" /
          WATER                    0.96
          COOLING-WATER            0.68
          REFRIGERANT              0.77
          VAPOR                    0.81
          ENERGY                   0.83
/
* Supply costs ----------
S_costs(S)  "supplies unitary costs (USD/kg)" /
          SULF-ACID                0.13
          ENZYME                   0.0164
          REP-ETHANOL              0.48
          REP-WATER                0
          CSL                      0.55
          MICROORG                 0
          CAL-HYDROX               0.19
          CATALYST                 2.5
          STEAM                    0
          AIR                      0
          ISOPARAFFIN              3.44
```

```
          YEAST-EXTRACT            4.5
          CO2                      0
/
* Stream costs ----------
ST_costs(ST)  "streams unitary costs (USD/kg) ** EXCLUDING FEEDSTOCKS **" /
          CHIPS                    10
          WHEY-CFG                 0
          LIPIDS                   0
          PULP                     0
          BLACK-LIQUOR             0
          HMW-LIGNIN               0
          GLUCOSE                  0
          LACTOSE                  0
          PROTEINS                 0
          ACETONE                  0
          BUTANOL                  0
          LAC-ACID                 0
          PLA                      0
          ISOPRENE                 0
          GAS-PURGE                0
          LIQ-PURGE                0
          SOL-PURGE                0
/
* Stream selling prices ----------
ST_prices(ST)  "streams unitary selling prices (USD/kg) ** EXCLUDING FEEDSTOCKS **" /
          CHIPS                    0
          WHEY-CFG                 0
          LIPIDS                   0
          PULP                     0
          BLACK-LIQUOR             0
          HMW-LIGNIN               0
          GLUCOSE                  0
          LACTOSE                  0
          PROTEINS                 0
          ACETONE                  0
          BUTANOL                  0
          LAC-ACID                 2.5
          PLA                      4.7
          ISOPRENE                 2.3
          GAS-PURGE                0
          LIQ-PURGE                0
          SOL-PURGE                0
/
* Feedstock costs ----------
C_F(F)  "feedstocks unitary costs (USD/kg)" /
          EUC-GRAND                0.0125
          WHEY                     0.5
/
Scalars
          M "Big M for restricting intermediaries binaries" /1000000000/
```

```
            CEPCI_adj "CEPCI adjustment for costs" /0.95538/
            X_min "Minimum flow for technologies" /1/
            base_salary "Operator base salary per year (USD)" /20000/
            r "Rate of Interest" /0.1/
            annualization_factor "Annualization Factor for Annualized Costs" /0.3/
  ;

* Key streams ----------
  X_keystream_1('FEEDSTOCK',g1)$(index_map_1('COND-LIG','EUC-GRAND',g1)) = 27777.79;
  X_keystream_1('FEEDSTOCK',g1)$(index_map_1('CENTRIF','WHEY',g1)) = 45131.82;
  X_keystream_2('CHIPS',g2)$(index_map_2('OSOLV','COND-LIG','EUC-GRAND',g2)) = 27777.79;
  X_keystream_2('CHIPS',g2)$(index_map_2('ST-EXPL','COND-LIG','EUC-GRAND',g2)) = 27777;
  X_keystream_2('WHEY-CFG',g2)$(index_map_2('ULTRAFIL','CENTRIF','WHEY',g2)) = 45122.92;
  X_keystream_3('PULP',g3)$(index_map_3('ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g3)) = 17178.2;
X_keystream_3('PULP',g3)$(index_map_3('ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g3)) = 11437.57;
X_keystream_4('GLUCOSE',g4)
$(index_map_4('PROD-LAC','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 27754.67;
  X_keystream_4('GLUCOSE',g4)
  $(index_map_4('PROD-LAC','ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g4)) = 49337.17;
  X_keystream_4('LACTOSE',g4)
  $(index_map_4('PROD-LAC','BLANK-3','ULTRAFIL','CENTRIF','WHEY',g4)) = 16085.47;
  X_keystream_4('GLUCOSE',g4)
  $(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 22900;
  * Inputs ----------
  INP_1('FEEDSTOCK',g1)$(index_map_1('COND-LIG','EUC-GRAND',g1)) = 27777.79;
  INP_1('FEEDSTOCK',g1)$(index_map_1('CENTRIF','WHEY',g1)) = 45131.82;
  INP_2('CHIPS',g2)$(index_map_2('OSOLV','COND-LIG','EUC-GRAND',g2)) = 27777.79;
  INP_2('CHIPS',g2)$(index_map_2('ST-EXPL','COND-LIG','EUC-GRAND',g2)) = 27777;
  INP_2('WHEY-CFG',g2)$(index_map_2('ULTRAFIL','CENTRIF','WHEY',g2)) = 45122.92;
  INP_3('PULP',g3)$(index_map_3('ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g3)) = 17178.2;
  INP_3('PULP',g3)$(index_map_3('ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g3)) = 11437.57;
  INP_4('GLUCOSE',g4)
  $(index_map_4('PROD-LAC','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 27754.67;
  INP_4('GLUCOSE',g4)
  $(index_map_4('PROD-LAC','ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g4)) = 49337.17;
  INP_4('LACTOSE',g4)
  $(index_map_4('PROD-LAC','BLANK-3','ULTRAFIL','CENTRIF','WHEY',g4)) = 16085.47;
  INP_4('GLUCOSE',g4)
  $(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 22900;
  * Outputs ----------
  OUT_1('CHIPS',g1)$(index_map_1('COND-LIG','EUC-GRAND',g1)) = 27777.79;
  OUT_1('WHEY-CFG',g1)$(index_map_1('CENTRIF','WHEY',g1)) = 45122.79;
  OUT_1('LIPIDS',g1)$(index_map_1('CENTRIF','WHEY',g1)) = 9.03;
  OUT_2('PULP',g2)$(index_map_2('OSOLV','COND-LIG','EUC-GRAND',g2)) = 1717.82;
  OUT_2('BLACK-LIQUOR',g2)$(index_map_2('OSOLV','COND-LIG','EUC-GRAND',g2)) = 1106.03;
  OUT_2('HMW-LIGNIN',g2)$(index_map_2('OSOLV','COND-LIG','EUC-GRAND',g2)) = 73.08;
  OUT_2('PULP',g2)$(index_map_2('ST-EXPL','COND-LIG','EUC-GRAND',g2)) = 11437.57;
  OUT_2('LIQ-PURGE',g2)$(index_map_2('ST-EXPL','COND-LIG','EUC-GRAND',g2)) = 18108.76;
  OUT_2('LACTOSE',g2)$(index_map_2('ULTRAFIL','CENTRIF','WHEY',g2)) = 16085.47;
  OUT_2('PROTEINS',g2)$(index_map_2('ULTRAFIL','CENTRIF','WHEY',g2)) = 29037.44;
```

```
  OUT_3('GLUCOSE',g3)
  $(index_map_3('ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g3)) = 27754.64;
  OUT_3('GLUCOSE',g3)
  $(index_map_3('ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g3)) = 49337.16;
  OUT_3('LIQ-PURGE',g3)
  $(index_map_3('ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g3)) = 10490.54;
  OUT_4('LAC-ACID',g4)
  $(index_map_4('PROD-LAC','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 2981.27;
OUT_4('LAC-ACID',g4)
  $(index_map_4('PROD-LAC','ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g4)) = 3946.26;
OUT_4('LAC-ACID',g4)
  $(index_map_4('PROD-LAC','BLANK-3','ULTRAFIL','CENTRIF','WHEY',g4)) = 1993.09;
  OUT_4('ISOPRENE',g4)
  $(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 1022.18;

  * Supplies ----------
  SUP_1(S,g1) = 0;
  SUP_2('REP-ETHANOL',g2)$(index_map_2('OSOLV','COND-LIG','EUC-GRAND',g2)) = 119.33;
  SUP_2('STEAM',g2)$(index_map_2('ST-EXPL','COND-LIG','EUC-GRAND',g2)) = 1769.33;
  SUP_3('ENZYME',g3)$(index_map_3('ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g3)) = 10574.68;
SUP_3('SULF-ACID',g3)$(index_map_3('ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g3)) = 1.82;
SUP_3('ENZYME',g3)
  $(index_map_3('ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g3)) = 21215.64;
  SUP_3('REP-WATER',g3)
  $(index_map_3('ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g3)) = 27173.52;
  SUP_3('SULF-ACID',g3)
  $(index_map_3('ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g3)) = 1;
  SUP_4('MICROORG',g4)
  $(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 109.92;
  SUP_4('ISOPARAFFIN',g4)
  $(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 260.39;

  * Offsite utilities ----------
  UTIL_1(U,g1) = 0;
  UTIL_2('WATER',g2)$(index_map_2('OSOLV','COND-LIG','EUC-GRAND',g2)) = 465.38;
  UTIL_2('VAPOR',g2)$(index_map_2('ST-EXPL','COND-LIG','EUC-GRAND',g2)) = 1769.33;
  UTIL_3('WATER',g3)$(index_map_3('ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g3)) = 10470.17;
UTIL_3('WATER',g3)$(index_map_3('ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g3)) = 21099.5;
UTIL_4('WATER',g4)
$(index_map_4('PROD-LAC','ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g4)) = 758.94;
UTIL_4('WATER',g4)
$(index_map_4('PROD-LAC','BLANK-3','ULTRAFIL','CENTRIF','WHEY',g4)) = 393.0922;
  UTIL_4('VAPOR',g4)
  $(index_map_4('PROD-LAC','BLANK-3','ULTRAFIL','CENTRIF','WHEY',g4)) = 27973.4552;
  UTIL_4('COOLING-WATER',g4)
  $(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 285350;
  UTIL_4('ENERGY',g4)
  $(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 350.17;
  UTIL_4('REFRIGERANT',g4)
  $(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 3521979.21;
```

```
  * Purchased utilities ----------
PUR_UTIL_1(PU,g1)   =   0;
PUR_UTIL_2(PU,g2)   =   0;
PUR_UTIL_3(PU,g3) = 0;
  PUR_UTIL_4('ENERGY',g4)
  $(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 405534;

* Equipment costs ----------
  C_E_1(g1)$(index_map_1('COND-LIG','EUC-GRAND',g1)) = 19000;
  C_E_1(g1)$(index_map_1('CENTRIF','WHEY',g1)) = 532900;
  C_E_2(g2)$(index_map_2('OSOLV','COND-LIG','EUC-GRAND',g2)) = 1941200;
  C_E_2(g2)$(index_map_2('ST-EXPL','COND-LIG','EUC-GRAND',g2)) = 783496;
  C_E_2(g2)$(index_map_2('ULTRAFIL','CENTRIF','WHEY',g2)) = 38847;
  C_E_3(g3)$(index_map_3('ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g3)) = 10133600;
  C_E_3(g3)$(index_map_3('ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g3)) = 12268500;
  C_E_4(g4)$(index_map_4('PROD-LAC','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 5068300;
C_E_4(g4)$(index_map_4('PROD-LAC','ENZ-HYD','ST-EXPL','COND-LIG','EUC-GRAND',g4)) = 4721937;
C_E_4(g4)$(index_map_4('PROD-LAC','BLANK-3','ULTRAFIL','CENTRIF','WHEY',g4)) = 4902800;
  C_E_4(g4)$(index_map_4('PROD-ISO','ENZ-HYD','OSOLV','COND-LIG','EUC-GRAND',g4)) = 6895400;

* Number of operations per types ----------
N_ops_tech('COND-LIG','sol') = 1;
  N_ops_tech('CENTRIF','flu') = 1;
  N_ops_tech('OSOLV','sol') = 1;
  N_ops_tech('OSOLV','sol_flu') = 1;
  N_ops_tech('OSOLV','flu') = 4;
  N_ops_tech('ST-EXPL','sol') = 1;
  N_ops_tech('ST-EXPL','sol_flu') = 1;
  N_ops_tech('ST-EXPL','flu') = 1;
  N_ops_tech('ULTRAFIL','sol_flu') = 1;
  N_ops_tech('ENZ-HYD','sol') = 1;
  N_ops_tech('ENZ-HYD','sol_flu') = 1;
  N_ops_tech('PROD-LAC','sol') = 2;
  N_ops_tech('PROD-LAC','flu') = 8;
 N_ops_tech('PROD-ISO','sol_flu') = 1;
```