Imperial College London

Vaccine Manufacturing Modelling Webinar

Developing Countries Vaccine Manufacturers Network (DCVMN)

13 December 2018

Zoltan Kis, Maria Papathanasiou, Cleo Kontoravdi and Nilay Shah
Centre for Process Systems Engineering
Chemical Engineering

Imperial College London

Outline

- **Vaccine Manufacturing Hub**
- Work Stream 1 (WS1): Improving existing processes and platforms
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies
- Modelling tools:
    - gPROMS (unit operation level);
    - SuperPro Designer (process flow sheet level);
    - GAMS (manufacturing and supply chain level).

Imperial College London

Outline

- Work Stream 1 (WS1): Improving existing processes and platforms

- Work Stream 2 (WS2): Design, development and implementation of new platform technologies

- Modelling tools:
    - gPROMS (unit operation level);
    - SuperPro Designer (process flow sheet level);
    - GAMS (supply chain level).

Imperial College London

WS1. Integrating life science and engineering to improve existing platforms

- LMIC partners have platforms for the production of vaccine formulations suitable for LMIC environment
- WS1 focuses on how best to exploit and further develop these to
  - enable mass vaccination campaigns
  - deal effectively with new outbreaks
- Objectives:
  - operational efficiency for cost reduction
  - rapid response of existing assets
  - end-to-end system design
- How? Whole process analysis and optimisation to address bottlenecks
  - In Life Sciences: host cell system or vector optimisation for improved productivity/quality
  - In Engineering: downstream separations, formulation and packaging

Imperial College London

WS1. Methodology

- Current capabilities
  - 90% of the LMIC vaccine production is attenuated and inactivated bacterial/virus vaccines
  - 10% mostly conjugates
- We have established a computational platform for modelling and optimising vaccine manufacturing processes to reduce costs
  - Use this to create parallel models that describe existing capabilities in LMIC partners and possible alternatives
- Apply whole process optimisation, system design and process intensification to improve operational flexibility and efficiency
- Process intensification has great promise for cost reduction and improvement of responsiveness in vaccine manufacturing

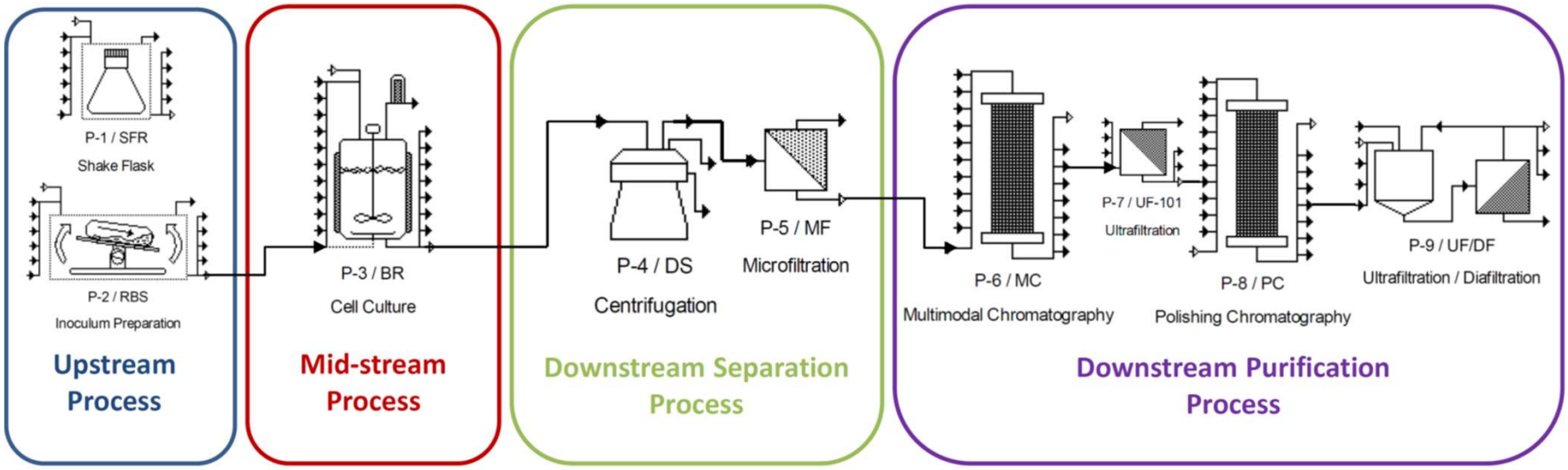
Imperial College London

WS1. Methodology

- Model, simulate and optimize the manufacturing and delivery processes at:
  I. unit operation level using gPROMS;
  II. process flow sheet level using SuperPro Designer,
  III. supply chain level using GAMS
- Perform stochastic sensitivity analysis to determine the input variables that have the highest potential for cost reduction, then further minimize costs by adjusting the high-impact variables.
- Enhance bioprocessing by:
  a) process intensification continuous processing strategies;
  b) de-bottlenecking;
  c) “process telescoping” combining several unit operations into one (e.g. charge and size based separation in one unit operation).
- Integration with experimental work, iterative

Imperial College London

WS1. Modular Design



- Advantages of a modular design:
  - Plug & Play
  - Standardization
  - Flexibility
  - Streamlined troubleshooting
  - Easy of training operators

Imperial College London

WS1. Rapid prototyping of novel downstream separation process concepts

- Explore two purification concepts for whole virus/bacteria, sub-unit and proteins:
  - “process telescoping” whereby several unit operations are combined into one (e.g. expanded bed affinity adsorption combining solids removal, capture and primary purification)
  - continuous operation (e.g. moving to continuous chromatography using simulated moving bed technology).
- Our key activities will involve high throughput experiments, models and big data analytics.
- Deliverable: Demonstration of new vaccine separation design concepts at lab scale

Imperial College London

WS1. Key drivers for continuous manufacturing

- Manufacturing capacity
- Reduced footprint, labour costs and CapEx
- Flexibility
- Speed to market
- Improved quality through the application of QbD & PAT

https://biopharma-asia.com/featured-article/challenges-development-continuous-processes-vaccines/

Imperial College London

WS1. Challenges

Design

- Design and control interactions
- Bioreactor design
- Downstream setup configuration

Control

- Complex operation profiles
- Complicated process models
- Unavailable measurements

Operation

- Optimal setup configuration
- Feasible operation
- Task coordination (scheduling)

Regulatory Bodies

- Enhanced process understanding
- Monitoring
- Global control

Imperial College
London

WS1. Requirements for a smooth transition

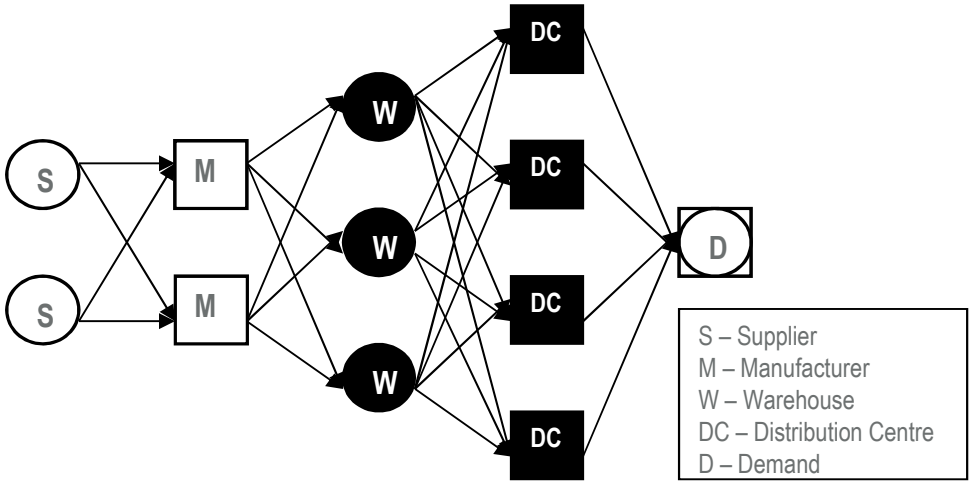
- ✓ Quality by Design (***QbD***)
- ✓ Thorough **understanding**
- ✓ Identification of **risks**/bottlenecks
- ✓ Process **monitoring**
- ✓ Global process **control**

…. **minimizing** experimental time & cost....

Imperial College London

WS1. Development of computational models for whole systems analysis

- Scaled up designs will be used to explore supply chain configuration:
  - centralised vs decentralised
  - shipment of bulk or fully-filled vaccines to clinics/local fill-finish plants
- What is a Supply Chain?
  - The alignment of firms that bring products or services to market
  - Linked by counter-current flow of material and information



Supply Chain Management
"The systemic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain, for the purposes of improving the long- term performance of the individual companies and the supply chain as a whole."

Imperial College London

WS1. Results

- State of the art conventional bacterial, viral and recombinant vaccine manufacturing processes have been reviewed.

- Calculated the batch volume for producing 1500 doses of the Hand, foot, and mouth disease (HFMD) vaccine in yeast at Dalian Hissen BioPharm Co., Ltd. (China).

Imperial College London

WS1. Ongoing case studies

- Analysing and improving vaccine manufacturing processes at MSD Wellcome TrustHillleman Laboratories Pvt. Ltd. (Hilleman Labs, India) for Recombinant cholera toxin B subunit and whole cell cholera vaccine production.
- Developing a model of the Vero cell based whole viral vaccine production and perform sensitivity analysis to quantify the relationships between CPPs and CQAs in support of QbD of this family of vaccine manufacturing processes. In collaboration with VaBiotech (Vietnam).
- We are engaging with additional vaccine developers and manufacturers in developing countries to simulate and optimize their processes (e.g. DCVMN meeting and webinars).

Imperial College London

Outline

- Work Stream 1 (WS1): Improving existing processes and platforms ✓

- Work Stream 2 (WS2): Design, development and implementation of new platform technologies

- Modelling tools:
  - gPROMS (unit operation level);
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).

Imperial College London

WS2. The 4 emerging vaccine platform technologies

RNA vaccines

- ~10 kb self-amplifying RNA
- regular mRNA
- produced by *in vitro* transcription

ADDomer vaccines

- adenovirus dodecahedron derived multimer
- 3.6 MDa VLP, 360 configurable epitopes, different epitope types
- insect cell-baculovirus expression

Humanized yeast vaccines

- glycoengineered yeast
- high-yield yeast expression of recombinant proteins

GMMA vaccines

- Generalized Modules for Membrane Antigen, outer membrane vesicles, 20-60 nm diameters
- configurable membrane proteins, bacterial expression

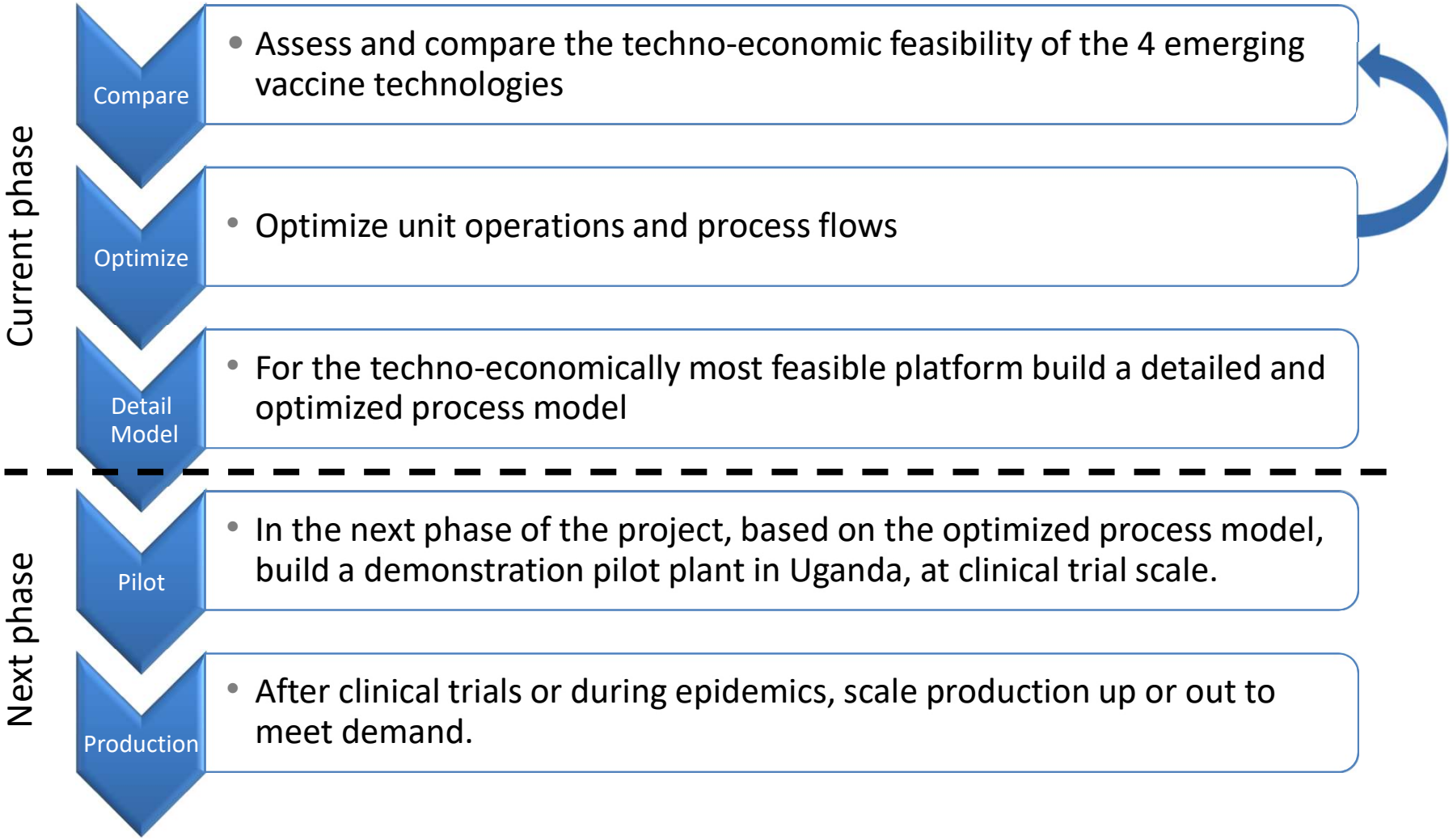
Imperial College London

WS2. Platform evaluation and improvement metrics

Speed	~100,000 vaccine doses, weeks after threat antigen identification
Cost	low cost, below 1 $/dose
Flexibility	on-demand production of a wide range of vaccine types (viral and bacterial)
Technological complexity	low technological complexity for implementation in developing countries
Technology readiness	mature technologies with licensed products & established manufacturing processes
Ease of scale-up or -out	highly scalable upstream and downstream processes
Thermo-stability of product	vaccines stable at 40°C for at least 6 months

Imperial College London

WS2. Workflow



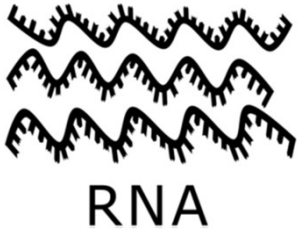
Imperial College London

WS2. Methodology

- To overcome data scarcity and uncertainty:
  i. Calculate parameters on a first-principle, bottom-up basis;
  ii. Estimate parameters based on similar processes and products by surveying the scientific literature and patent databases, and by interviewing experts;
  iii. Model, simulate and optimize the manufacturing processes at unit operation level (using gPROMS) and process flow sheet level (using SuperPro Designer, Aspen Batch Process Developer), at supply chain level (using GAMS)
- Perform stochastic sensitivity analysis to determine the input variables that have the highest potential for cost reduction, then further minimize costs by adjusting the high-impact variables.
- Enhance bioprocessing by: (a) evaluating process intensification continuous processing strategies; (b) de-bottlenecking; (c) “process telescoping” combining several unit operations into one

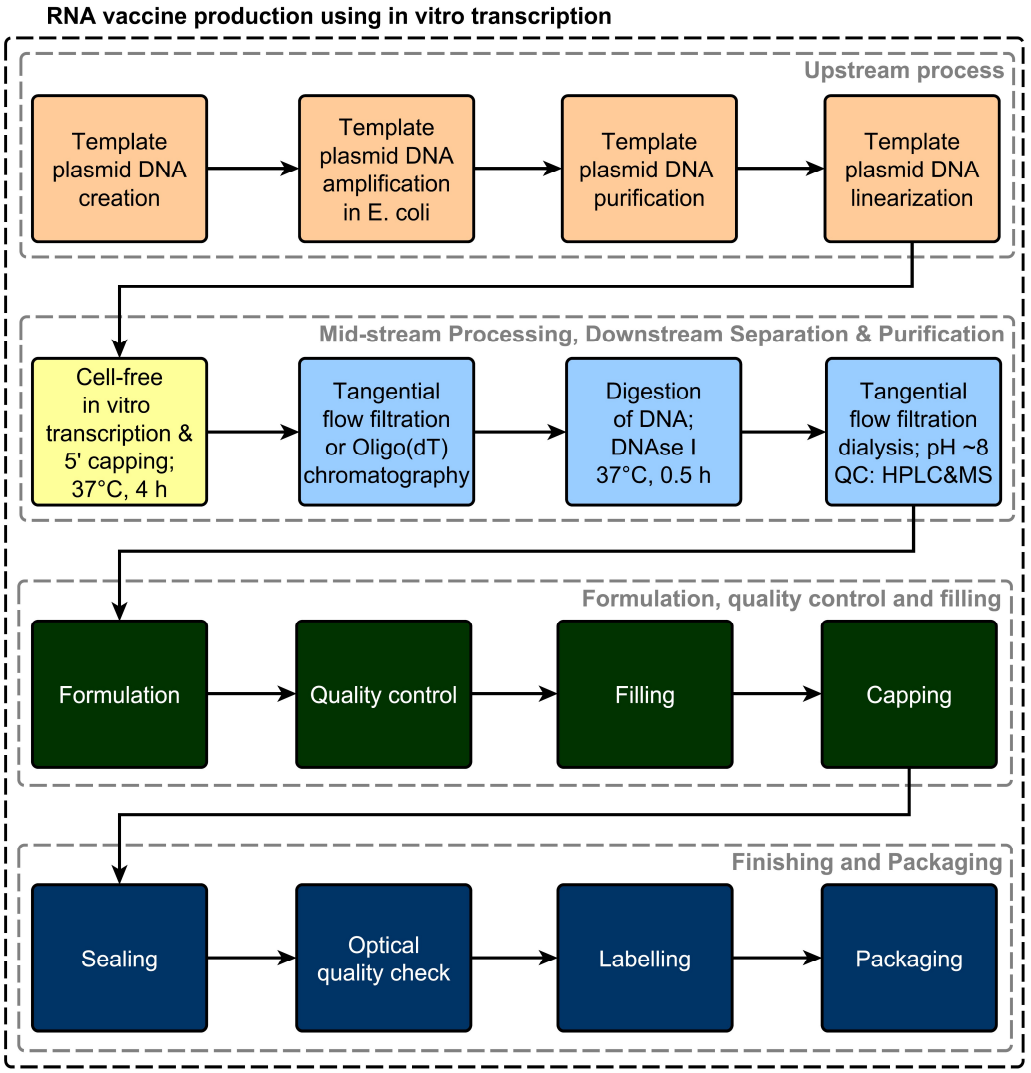
Imperial College London

WS2. RNA platform results



- ~10 kb self-amplifying RNA
- produced by *in vitro* transcription
- cell-free product
- Co-transcriptional 5’ capping, ARCA

Cost per dose at 20 µg/dose: 0.72 USD/dose

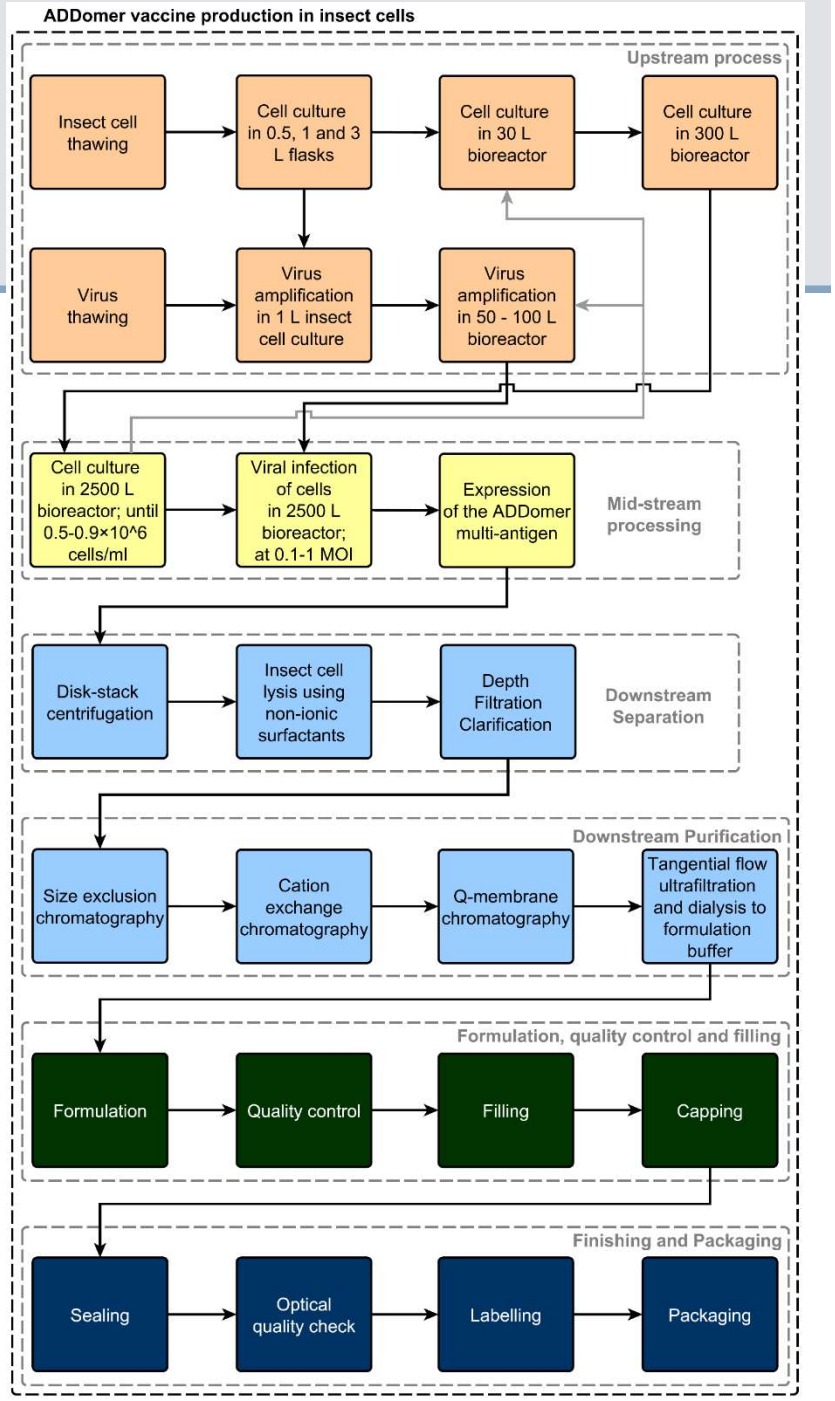


Imperial College London

WS2. ADDomer platform results

- adenovirus dodecahedron derived multimer
- 3.6 MDa VLP, 360 configurable epitopes, different epitope types
- insect cell-baculovirus expression
- intracellular

ADDomer

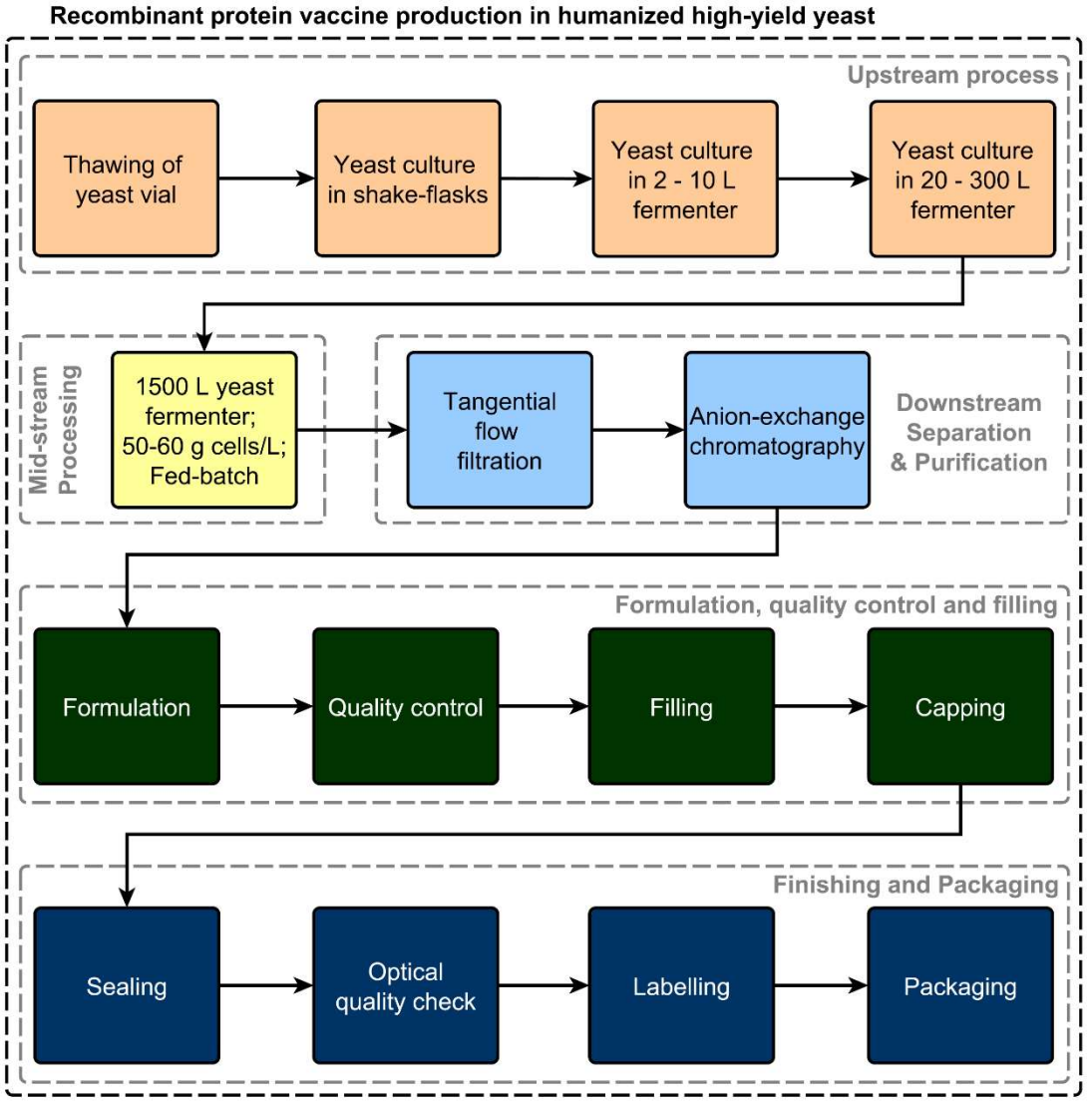


Imperial College London

WS2. Humanized yeast platform results

- glycoengineered yeast
- high-yield yeast expression of recombinant proteins
- extracellular

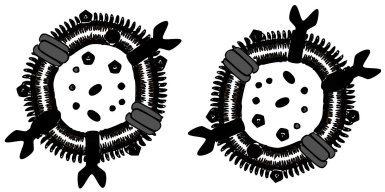
h Yeast



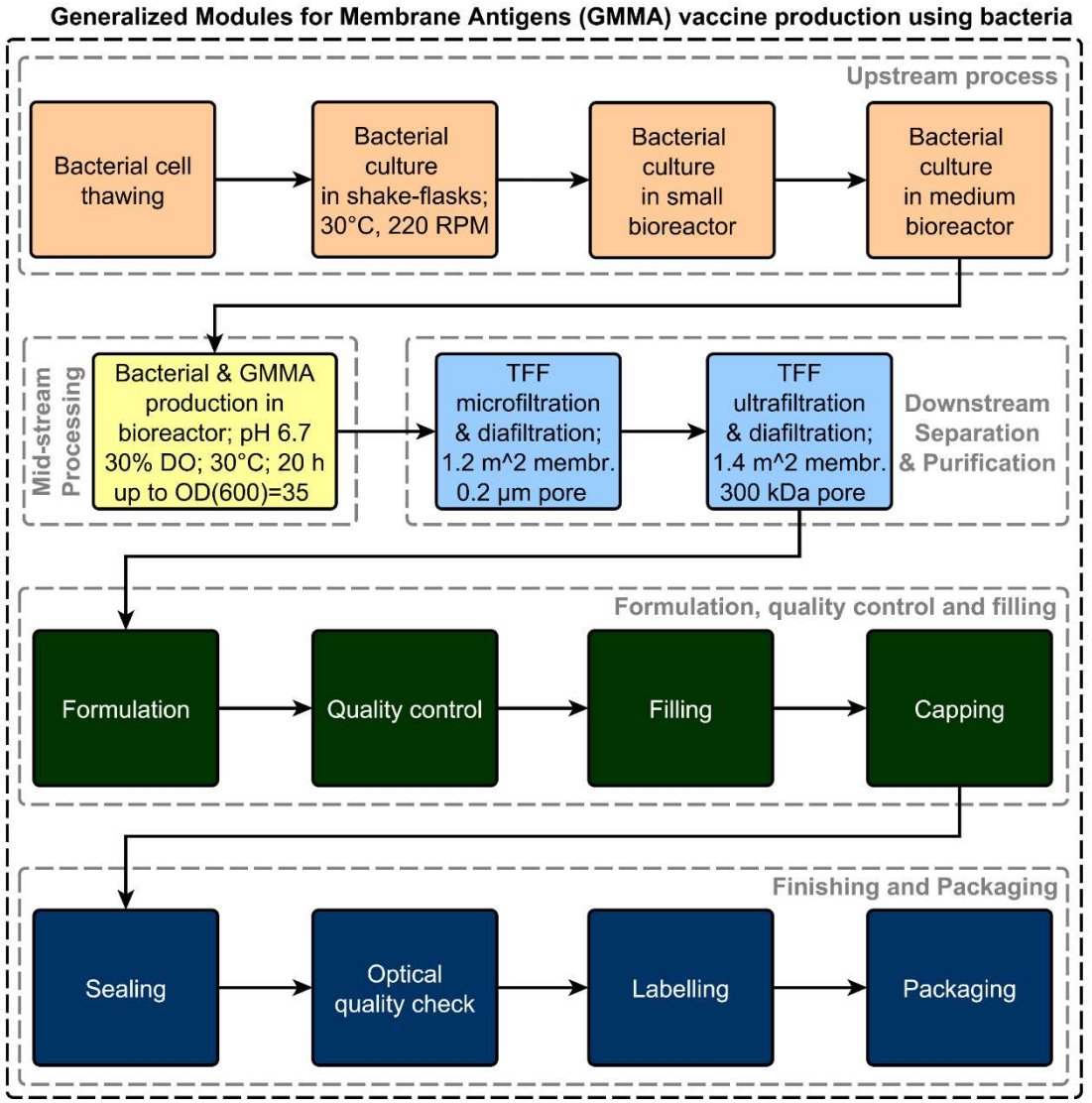
Imperial College London

WS2. GMMA platform results

- Generalized Modules for Membrane Antigen, outer membrane vesicles, 20-60 nm diameters
- configurable membrane proteins, bacterial expression
- extracellular



GMMA



Imperial College London

WS2. Feasibility and risk assessment results

Table 1. Feasibility and risk assessment of the 4 emerging platform technologies

Metric \ Platform [a]		Yeast platform	ADDomer platform	GMMA platform	RNA platform
1	Technology readiness	2	3	5	4
2	Technological complexity	3	1	5	2
3	Ease of scale-up and -out	4	2	5	3
4	Flexibility [b]	3	3	2	4
5	Thermo-stability of product	3	5	3	2
6	Speed of response	1	4	2	5
Sum: overall feasibility and risk estimate [c]		16	18	22	20

[a] Yeast platform - Humanised, high-yield yeast platform for recombinant vaccine manufacturing; ADDomer platform - Insect cell-baculovirus platform for recombinant vaccine manufacturing; GMMA platform - Outer membrane vesicle vaccines, Generalized Modules for Membrane Antigen vaccine manufacturing; RNA platform - RNA vaccine manufacturing.

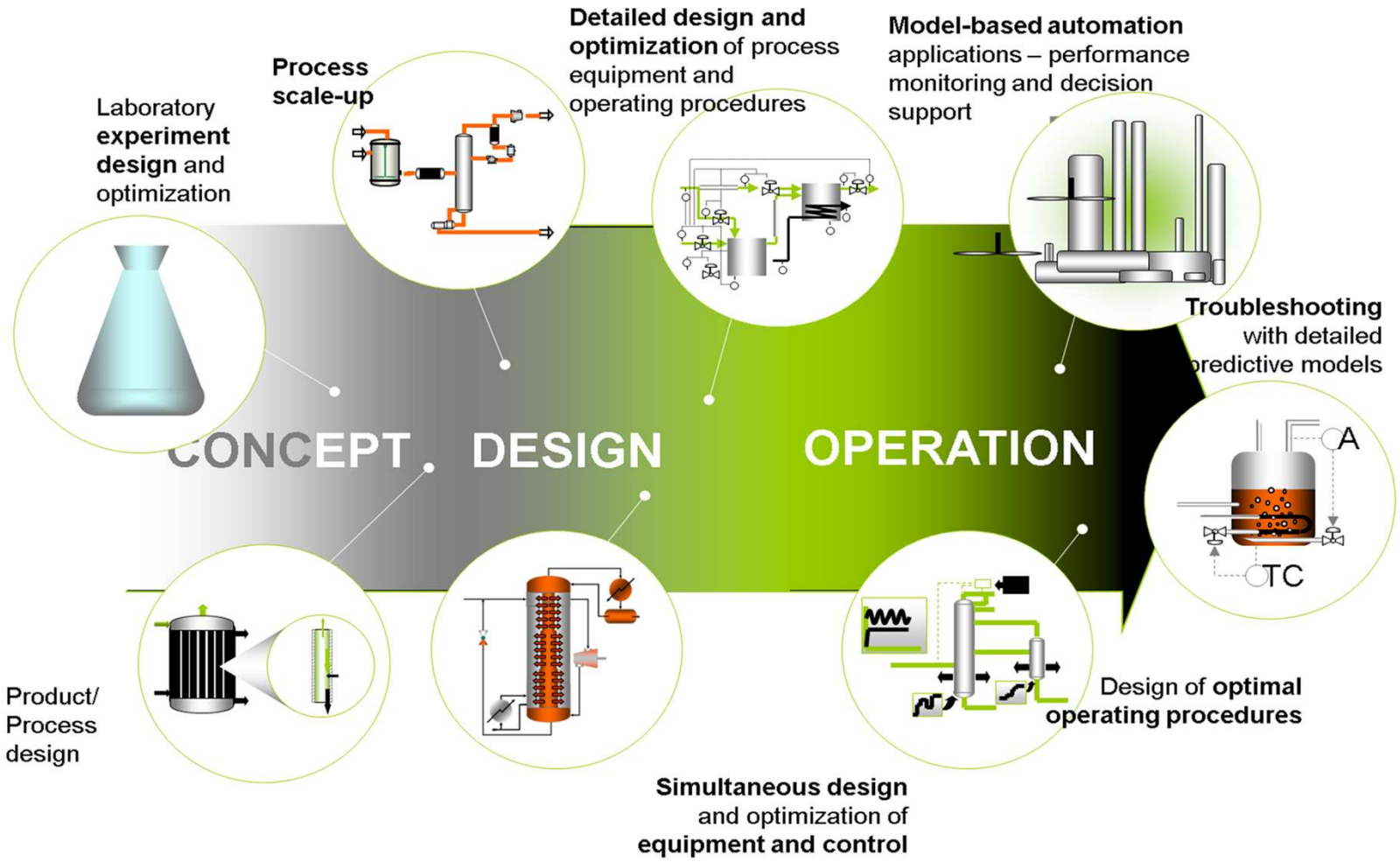
[b] Universal applicability for the manufacturing of a wide range of vaccines.

[c] The overall feasibility and risk estimate was calculated by summing up the values for each metric per technology.

Zoltán Kis, Robin Shattock, Nilay Shah, Cleo Kontoravdi. Emerging technologies for low-cost, rapid vaccine manufacture. Biotechnology Journal. Accepted. Oct 2018.

Imperial College London

Process Systems Engineering – lifecycle view



Imperial College London

WS2. Outcomes

Key outcomes from this workstream include:

- A screening methodology which can identify which platform is best suited to a particular vaccine
- Conceptual and detailed demonstration industrial process designs/blueprints which build on lab data
- Actual physical demonstrations of new manufacturing concepts with emerging data
- Application of these to demonstrate the benefits of novel platforms and approaches as an evidence base for regulators, healthcare providers and manufacturers.

Imperial College
London

WS2. Future tasks

Task 2.3: Application of detailed demonstration design methodology to up to 6 shortlisted demonstration process concepts, leading to:

(a) detailed designs and

(b) more accurate KPIs including cost,

which will be used to select the demonstration projects for physical deployment.

Task 2.4: Development of physical demonstrators - up to 6 scale appropriate physical demonstrators used to:

- advance the knowledge around industrialisation of the new platforms,
- evaluate actual performance in an industrial setting,
- feed back information to the model-based design activity to support analysis and design of full commercial scale processes.

Imperial College London

Outline

- Work Stream 1 (WS1): Improving existing processes and platforms ✓
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies ✓
- Modelling tools:
  - gPROMS (unit operation level);
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).

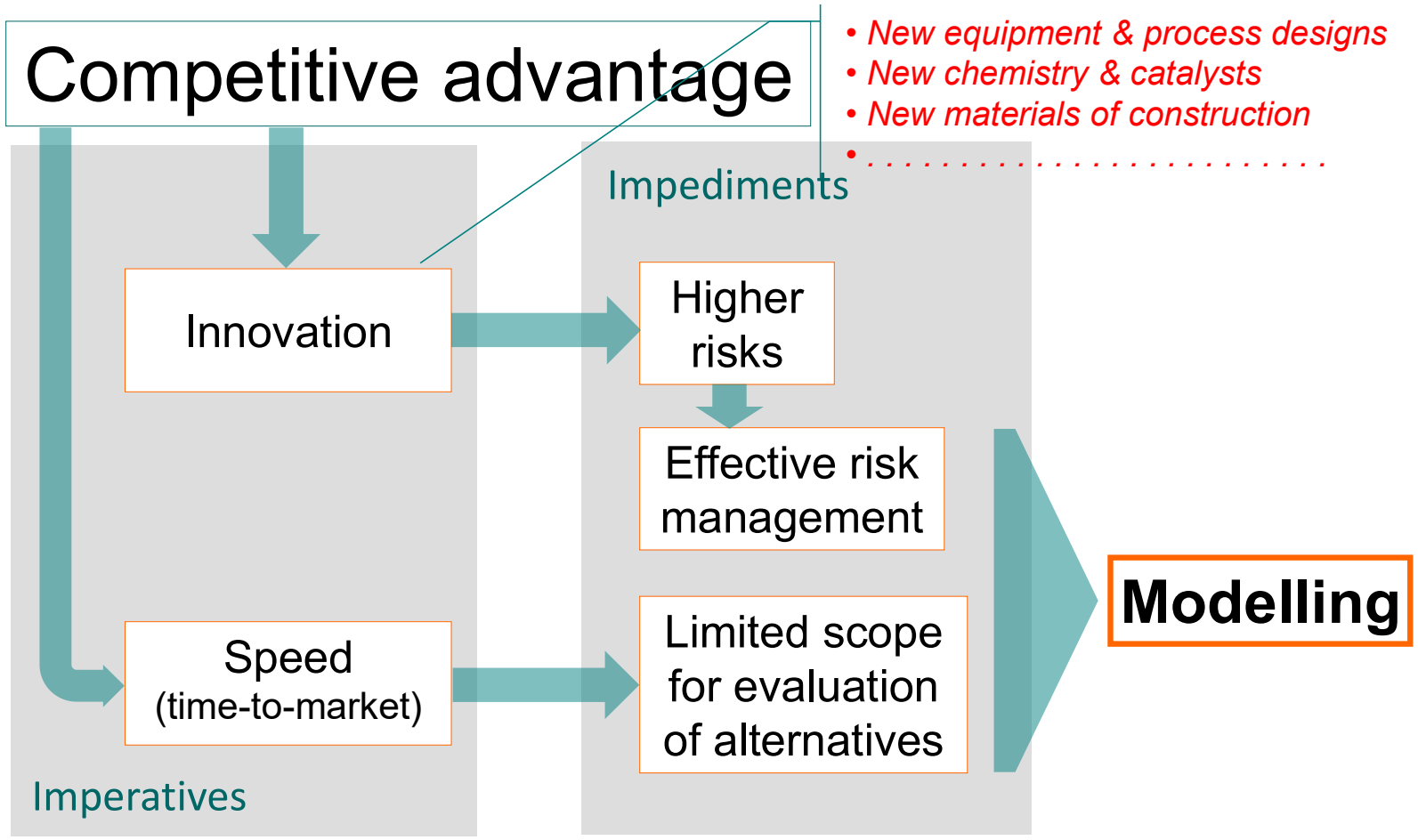
Imperial College London

Methodology

- We have established a computational platform for modelling and optimising vaccine manufacturing processes to reduce costs
  - Use this to create parallel models that describe existing capabilities in LMIC partners and possible alternatives
- Apply whole process optimisation, system design and process intensification to improve operational flexibility and efficiency
- Process intensification has great promise for cost reduction and improvement of responsiveness in vaccine manufacturing

Imperial College London

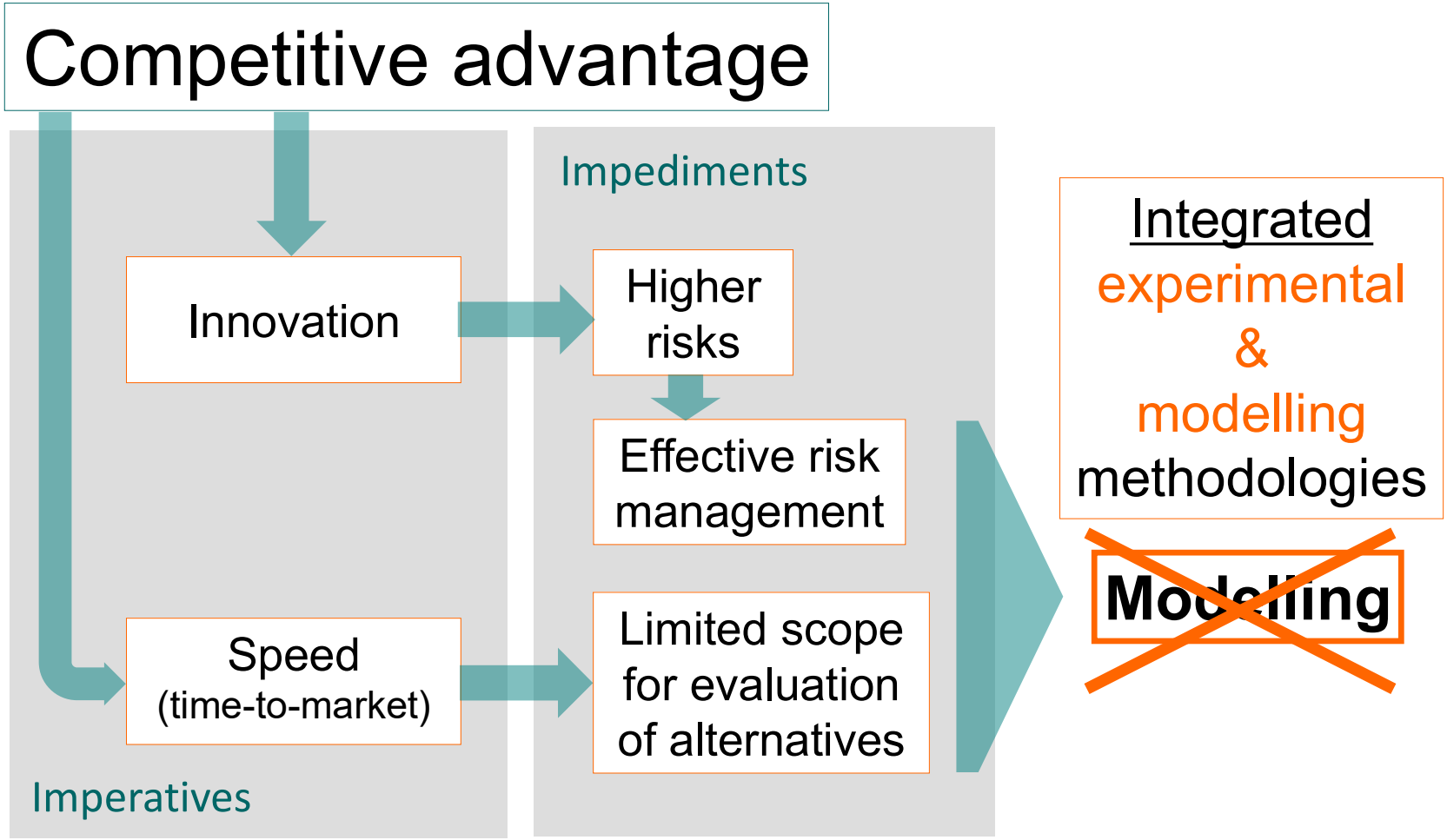
Why model?



(Costas Pantelides, PSE Ltd)

Imperial College London

Innovation, risk and modelling – models and data not models or data



Imperial College
London

Application of integrated approach for improving primary production

- Employ high-level models of traditional upstream systems (bacteria, yeast and animal cells) and our emerging platform
- Optimise using process mapping, bottleneck identification and process intensification, building on work in biologics manufacturing .
- Also identify raw materials and suitable alternatives available locally and/or at lower cost.
- Deliverable: Demonstration of benefits of integrated approach on primary production systems

Imperial College London

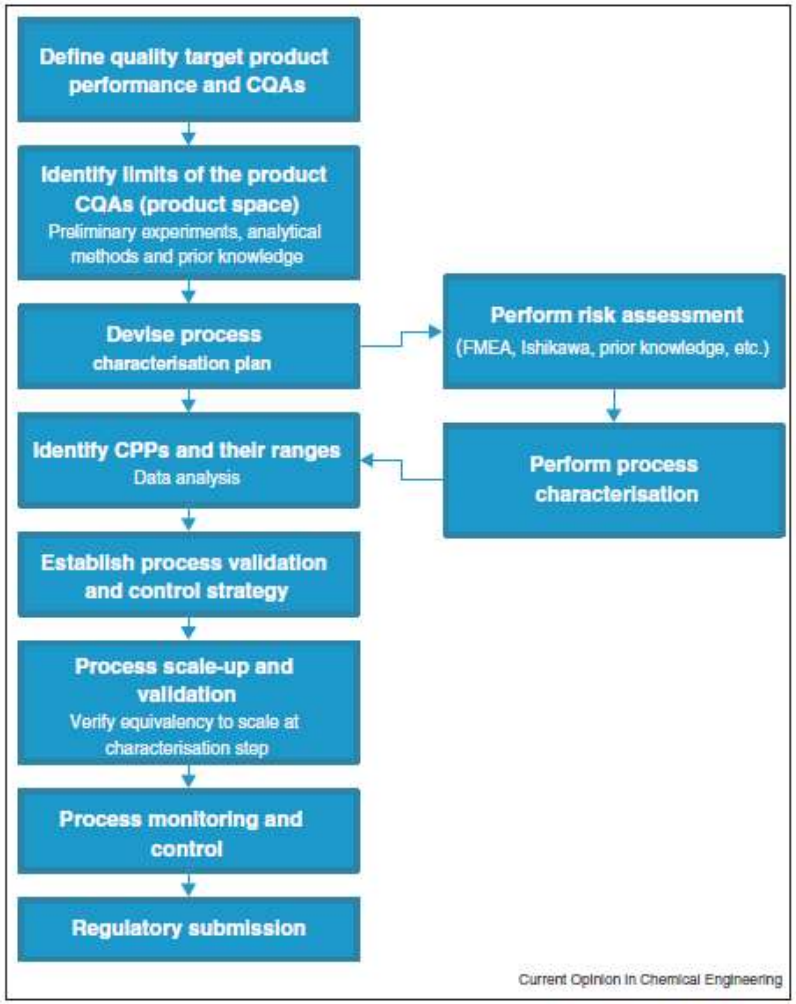
Development of computational models for whole systems analysis

- Multi-scale modelling of biological processes through to unit operation and whole value chains will be used for system analysis, design and manufacturing operation optimisation
- How do parameters characterising single unit performance e.g. titre, purity, recovery, formulation recipe influence whole system metrics e.g. cost per dose, lead times?
  - Identify priority areas of study

Imperial College London

Modelling for process optimisation

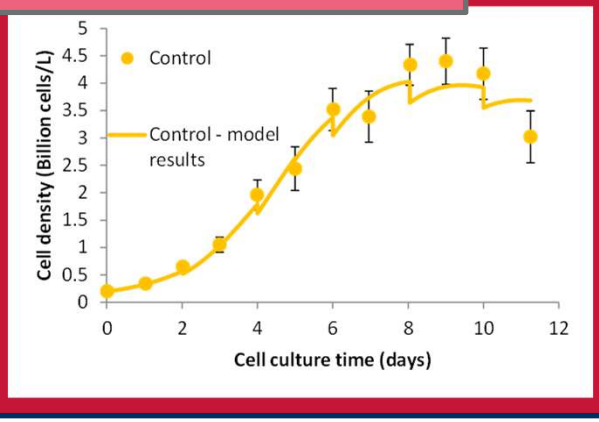
Following Quality by Design principle for increasing process understanding

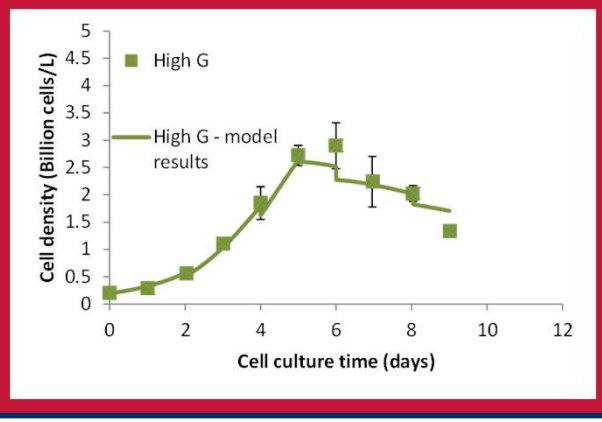


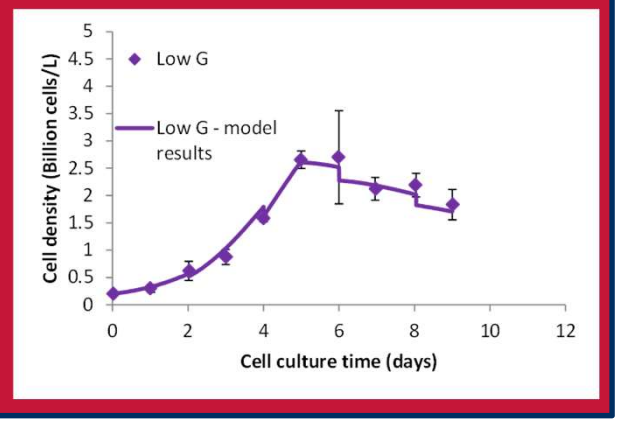
Imperial College London

Development of reliable process models

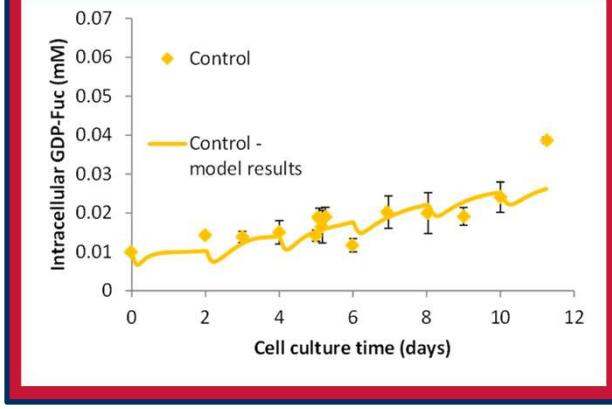
Cell Culture Dynamics

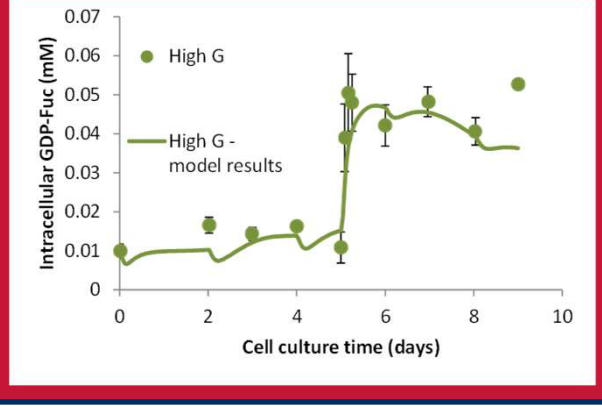


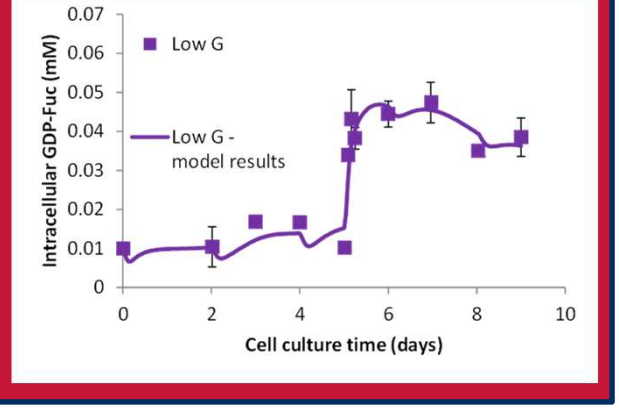




Intracellular kinetic model



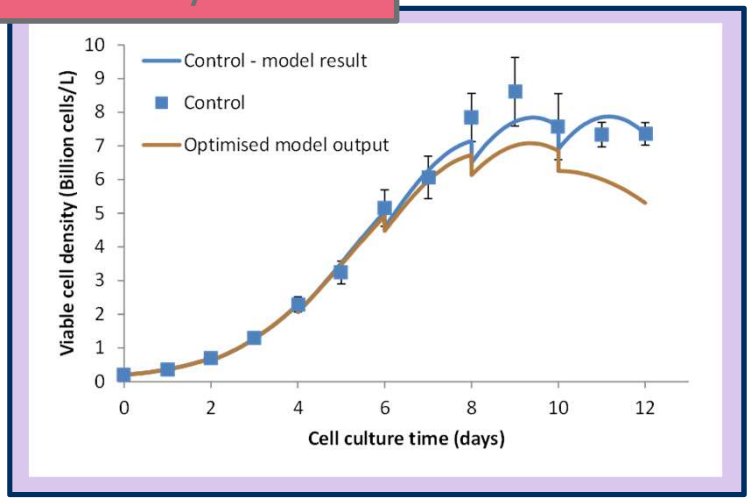




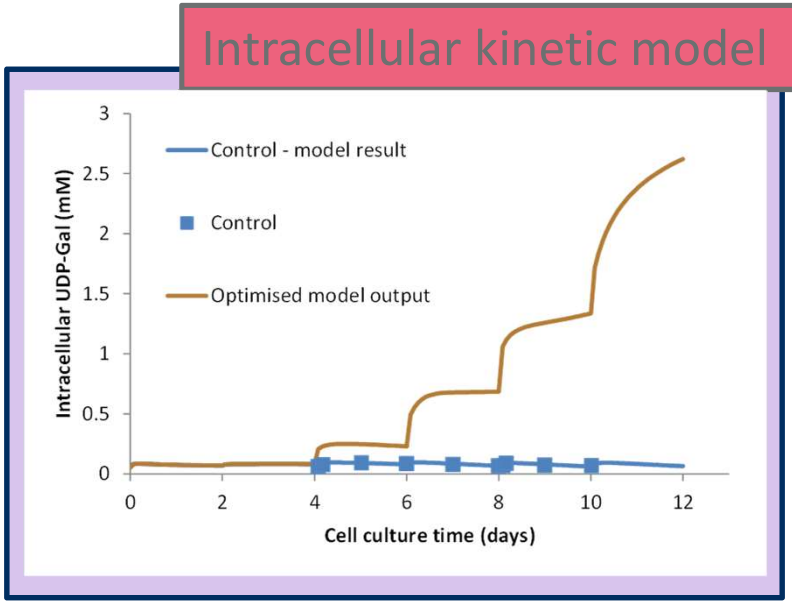
Imperial College London

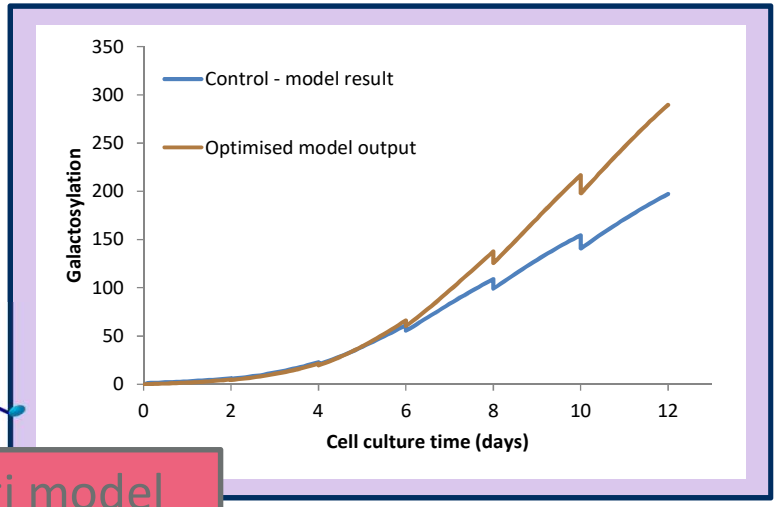
In silico optimisation

Cell Culture Dynamics



Intracellular kinetic model

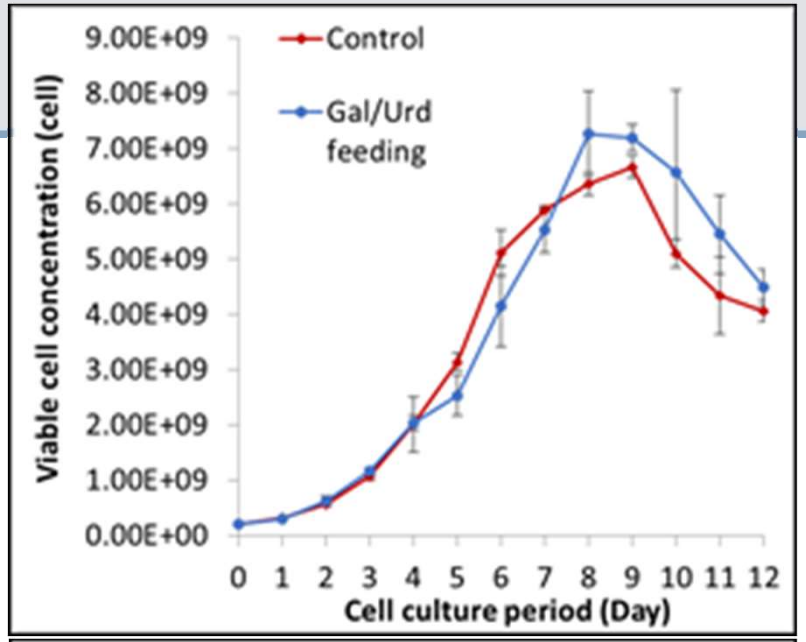


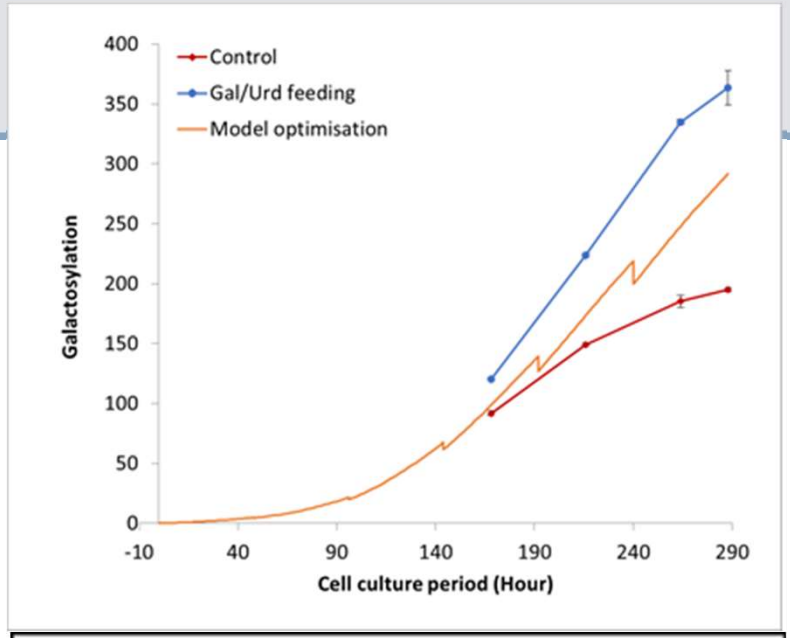


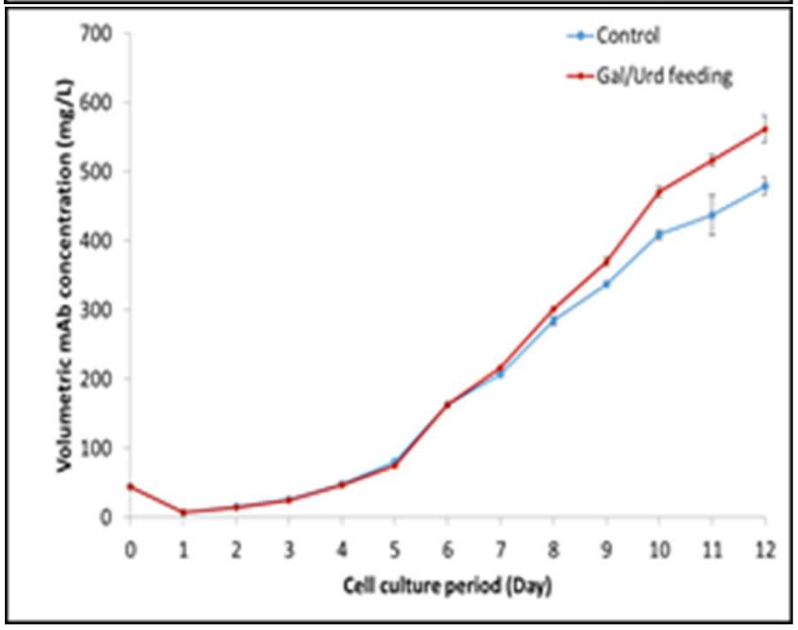
Golgi model

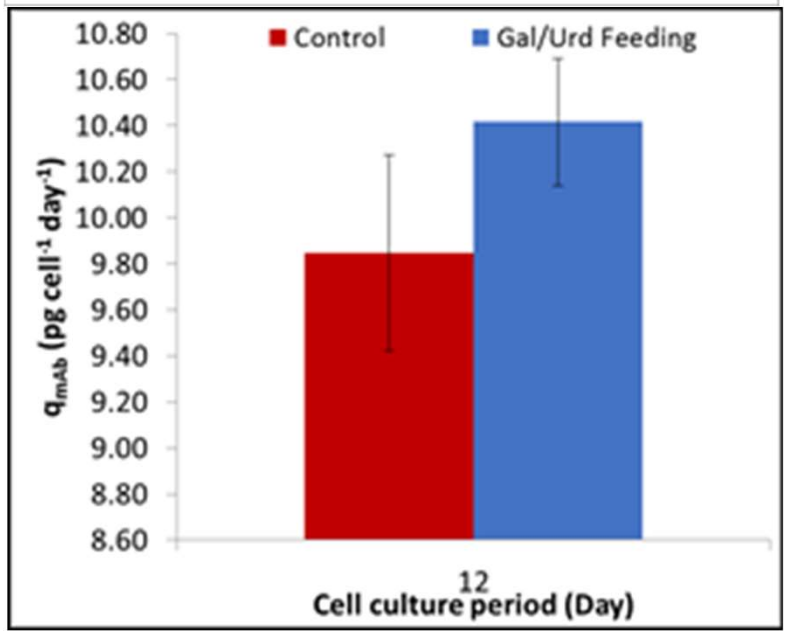
Imperial College London

Experimental results of optimal operating conditions



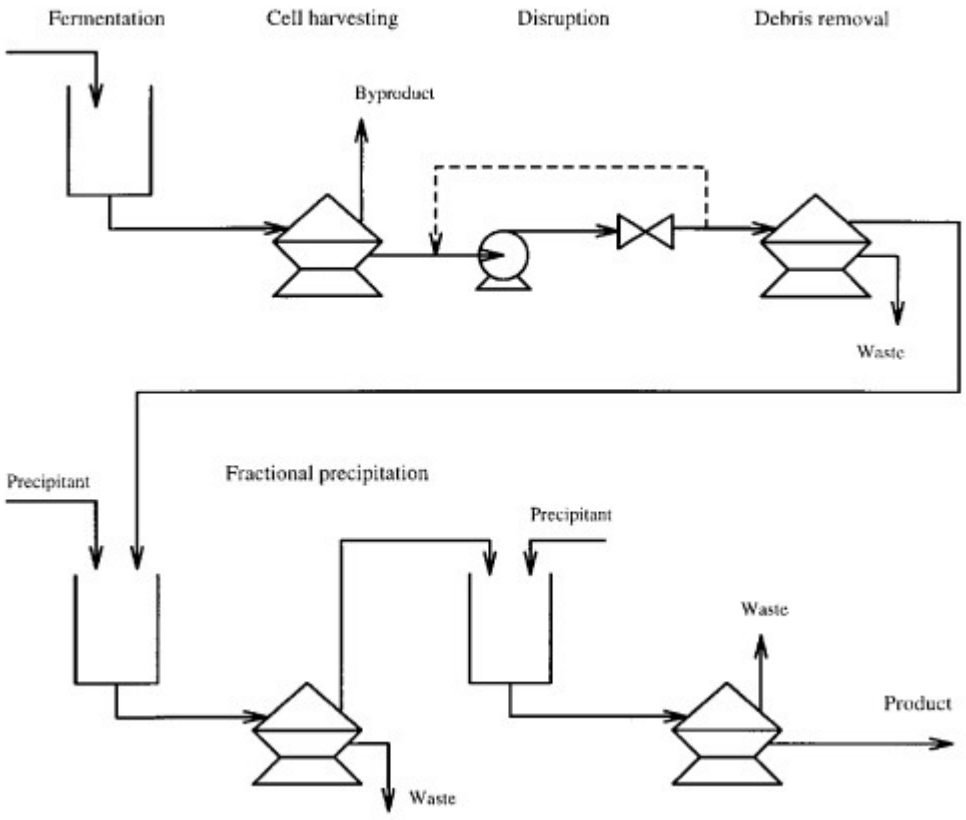






Imperial College
London

Whole process analysis: intracellular product



Imperial College
London

Initial process design and dynamic optimisation

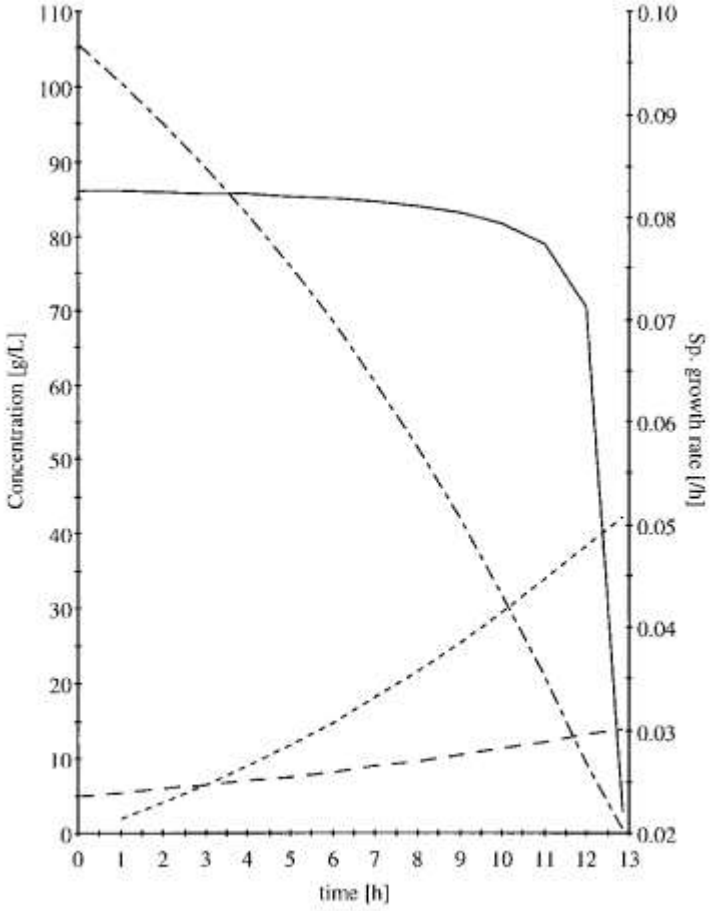


Figure 3. Optimal fermenter profiles, for problem 1 (Table 1).---- P; - . - . - S; - - - X; ——— μ.

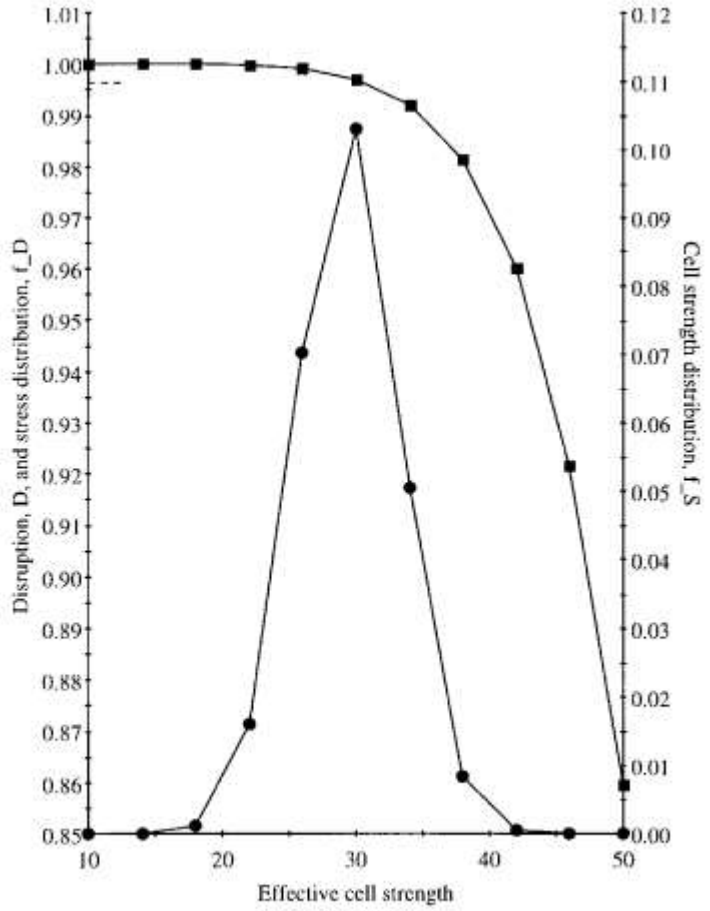


Figure 5. Stress and cell strength distribution in homogenizer, problem 1. - - - - D; ■ f-D; ● f-s.

Imperial College
London

Simultaneous design of key variables

Table 2. Optimal design for batch fermentation, maximizing profit, purity $\geq$ 0.13.

Fermenter	S_0	101.04
	X_0	5
	$V\ (= V_0)$	800
	N	1
	t_{ferm}	11.888
Cell harvesting centrifuge	F_{feed}	110
	ω	10000
	t_{hcent}	7.273
Homogenizer	F	60
	P	55
	t_{hom}	13.333
Centrifuge 1	F_{feed}	80
	ω	7234.3
	t_{cent1}	10
Precipitation stage 1	$M_{salt,in}$	0
	M_{salt}	32000
Centrifuge 2	F_{feed}	80
	ω	6500
	t_{cent2}	9.5
Precipitation stage 2	$M_{salt,in}$	32000
	M_{salt}	28000
Centrifuge 3	F_{feed}	75
	ω	9500
	t_{cent3}	10.397
Overall cycle time	t_{cyc}	13.333
Process performance	profit	48311
	purity	0.1299
	recovery	0.7585

Table 3. Optimal design for batch fermentation, maximizing purity.

Fermenter	S_0	107.61
	X_0	5
	$V\ (= V_0)$	800
	N	1
	t_{ferm}	11.655
Cell harvesting centrifuge	F_{feed}	115.85
	ω	9042.3
	t_{hcent}	6.905
Homogenizer	F	60
	P	80
	t_{hom}	13.333
Centrifuge 1	F_{feed}	60
	ω	2274.8
	t_{cent1}	13.333
Precipitation stage 1	$M_{salt,in}$	0
	M_{salt}	31114
Centrifuge 2	F_{feed}	70.376
	ω	7391.8
	t_{cent2}	10.799
Precipitation stage 2	$M_{salt,in}$	31114
	M_{salt}	18945
Centrifuge 3	F_{feed}	60
	ω	8000
	t_{cent3}	12.033
Overall cycle time	t_{cyc}	15.946
Process performance	profit	12505
	purity	0.1395
	recovery	0.7191

Imperial College
London

From design to operations

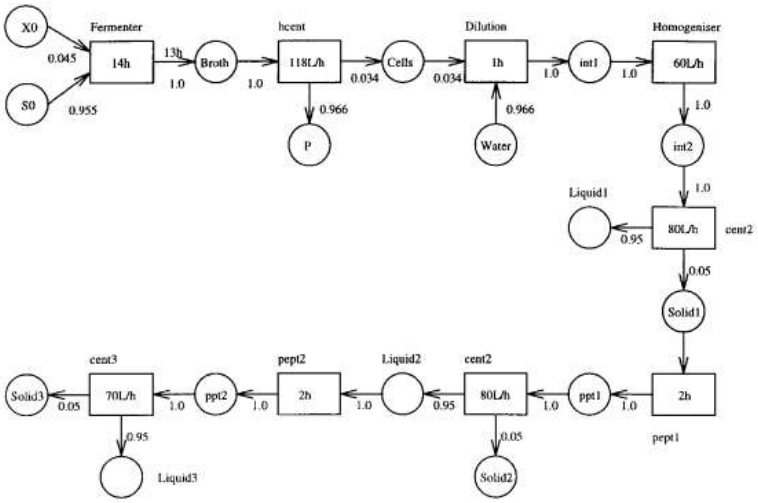


Figure 1. State task network for the example problem.

Table 1. Optimal storage tank capacities.

State	Tank capacity (l)
feed and product states	not sized
broth	800
cells	24.48
dcells	740
int1	125
liquid1	684
ppt1	760
liquid2	684
ppt2	652.34

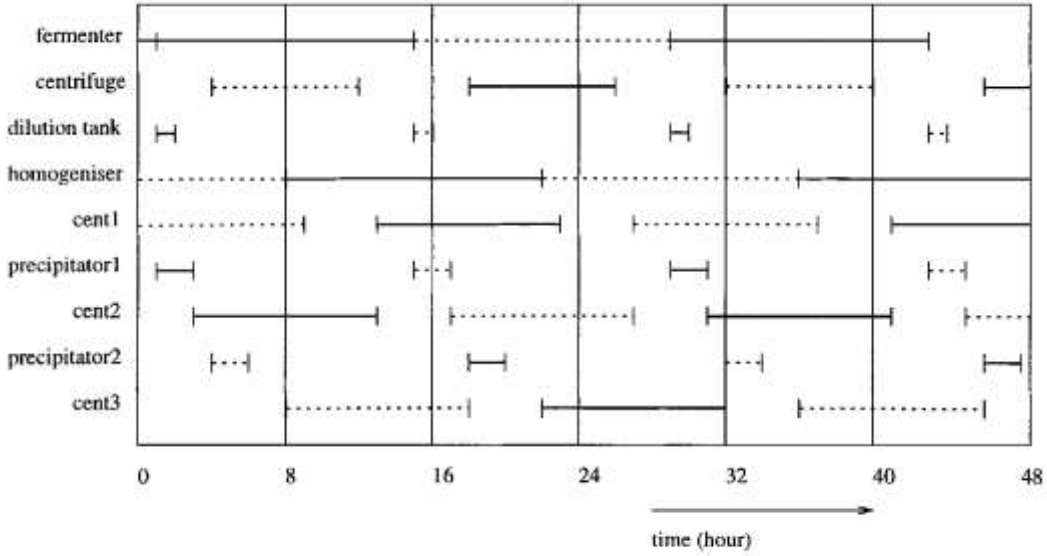
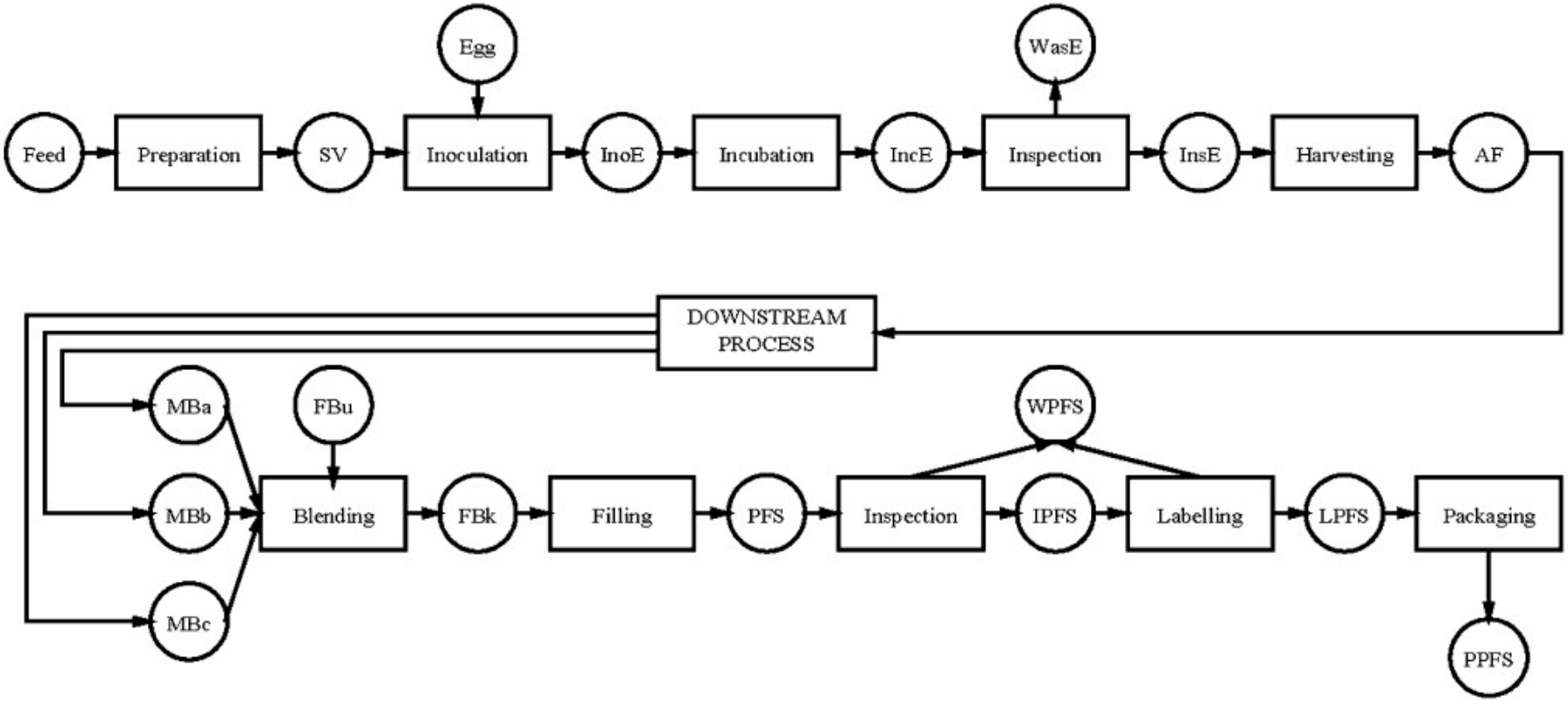


Figure 4. Cyclic schedule for one batch, as above, showing three batches.

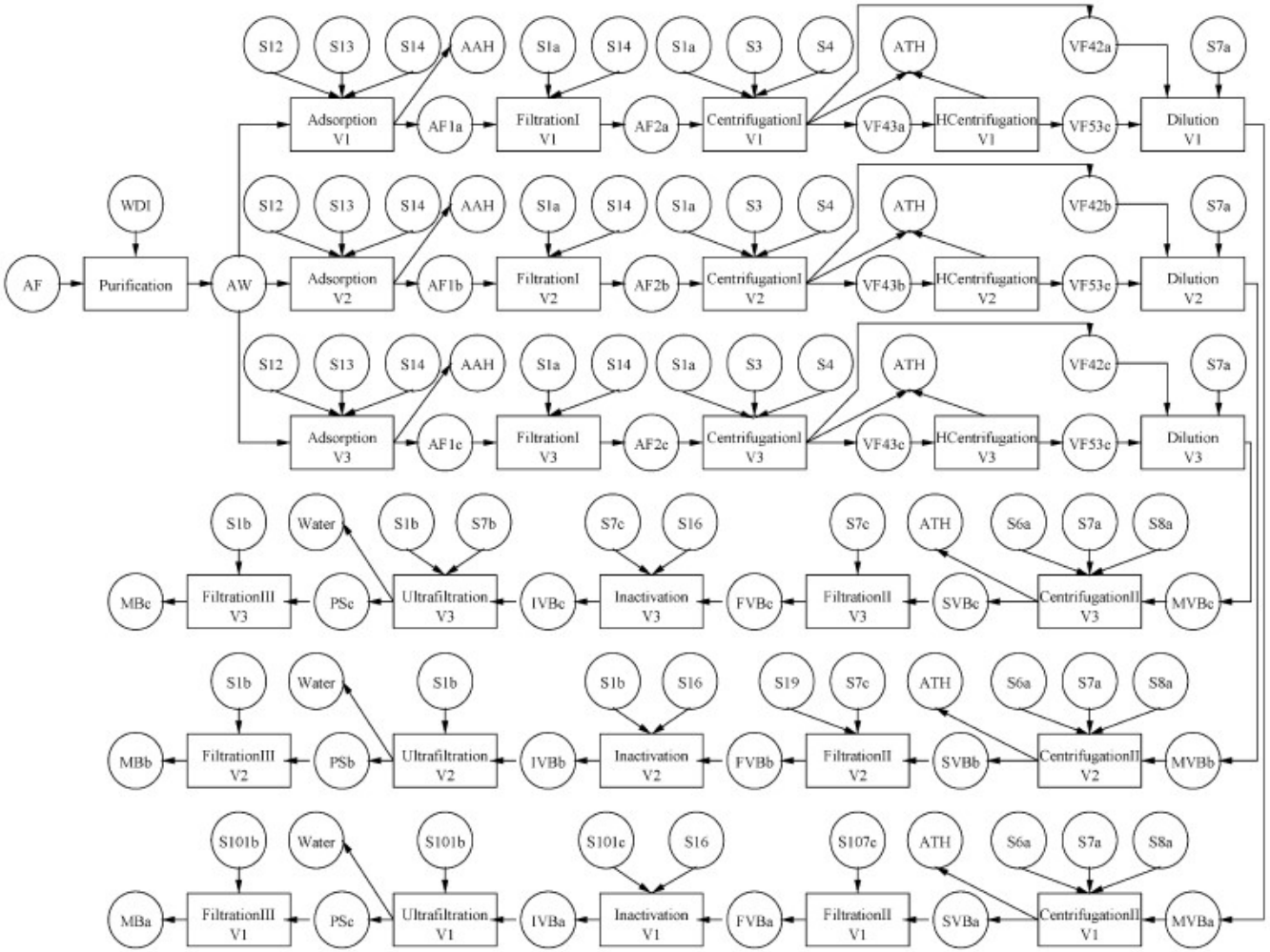
Imperial College
London

Debottlenecking of influenza vaccine process



Imperial College
London

Debottlenecking model

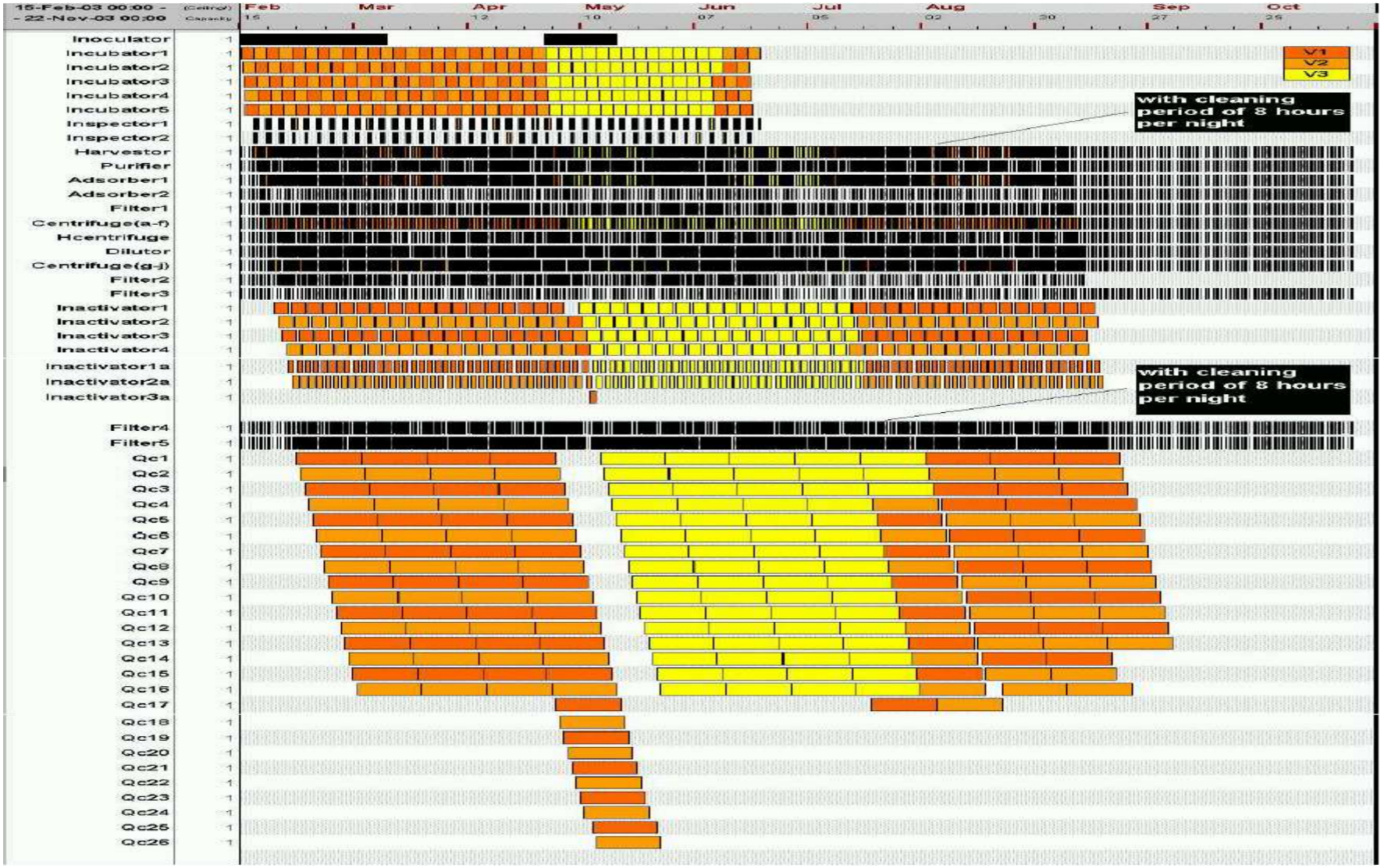


Imperial College London

Debottlenecking

Table 2. Comparisons of different methods debottlenecking strategies.

Case	Batches obtained (% change)	Bottleneck	Customer service level
A	67	Incubation	93%
B	93 (39%)	Incubation	100%
C	98 (46%)	Inactivation	100%
D	98 (46%)	Incubation	100%



Imperial College London

Incorporation of uncertainty

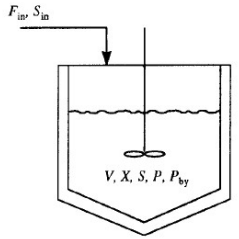


Figure 1: The fermenter tank

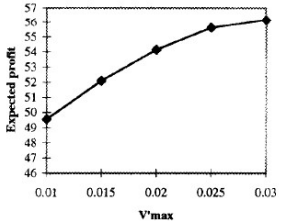


Figure 2: Expected profit vs V'_{max}

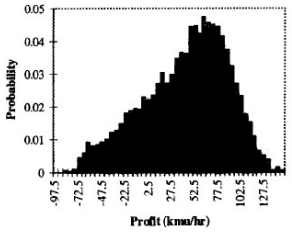


Figure 3: Profit distribution for nominal case

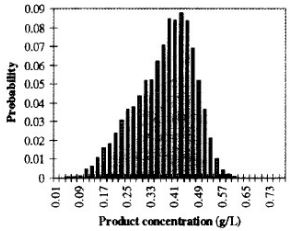


Figure 4: Product concentration distribution for nominal case

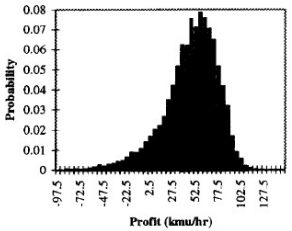


Figure 5: Profit distribution for robust case

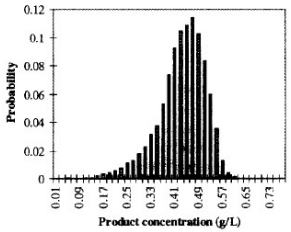
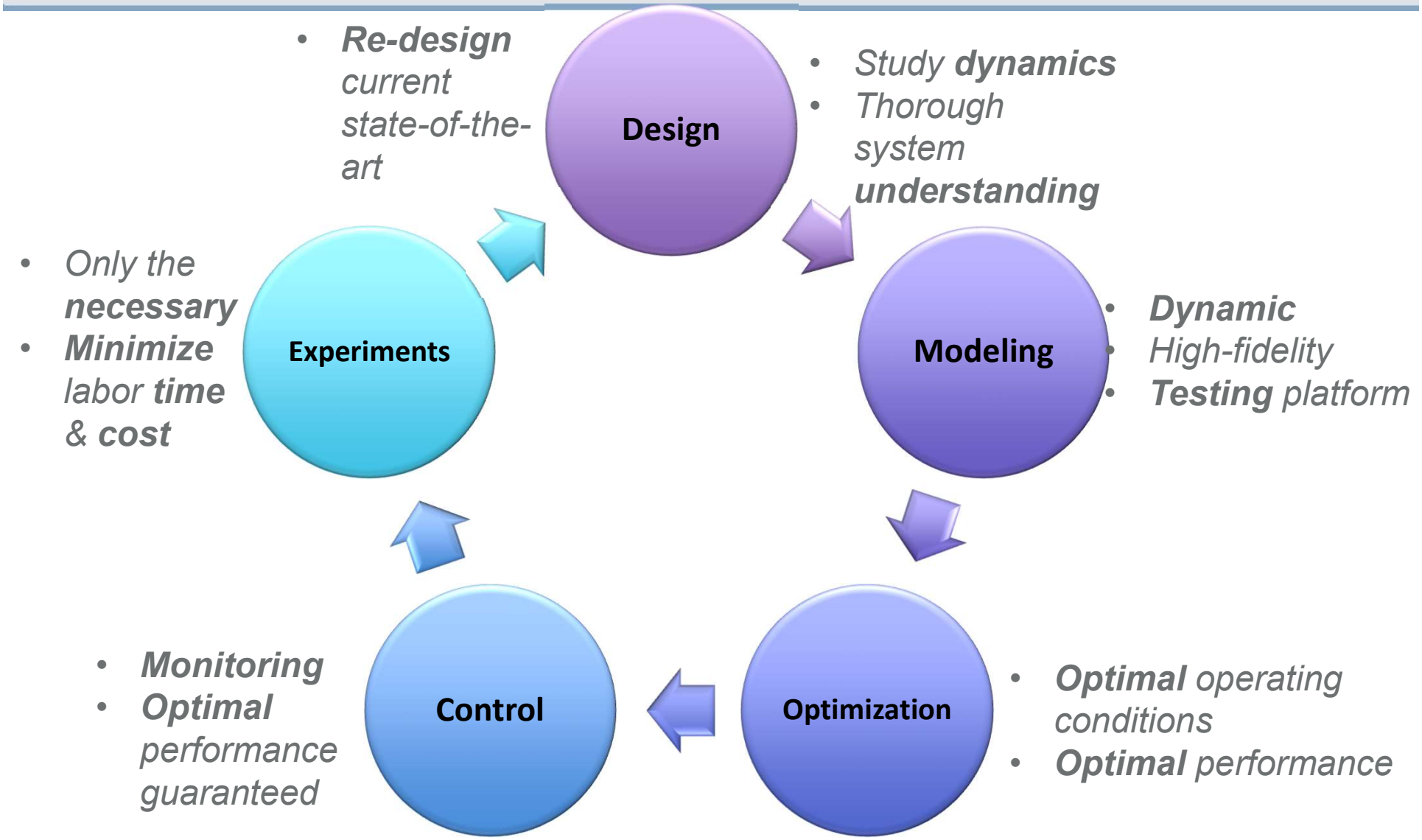


Figure 6: Product concentration distribution for robust case

Imperial College London

The aid of computational tools



Imperial College
London

Development of computational models for whole systems analysis

- Multi-scale modelling of biological processes through to unit operation and whole value chains will be used for system analysis, design and manufacturing operation optimisation
- How do parameters characterising single unit performance e.g. titre, purity, recovery, formulation recipe influence whole system metrics e.g. cost per dose, lead times?
  - Identify priority areas of study

Imperial College London

Outline

- Work Stream 1 (WS1): Improving existing processes and platforms ✓
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies ✓
- Modelling tools:
  - gPROMS (unit operation level);
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).

Imperial College London

Introduction to gPROMS

Describing new processes and unit operations

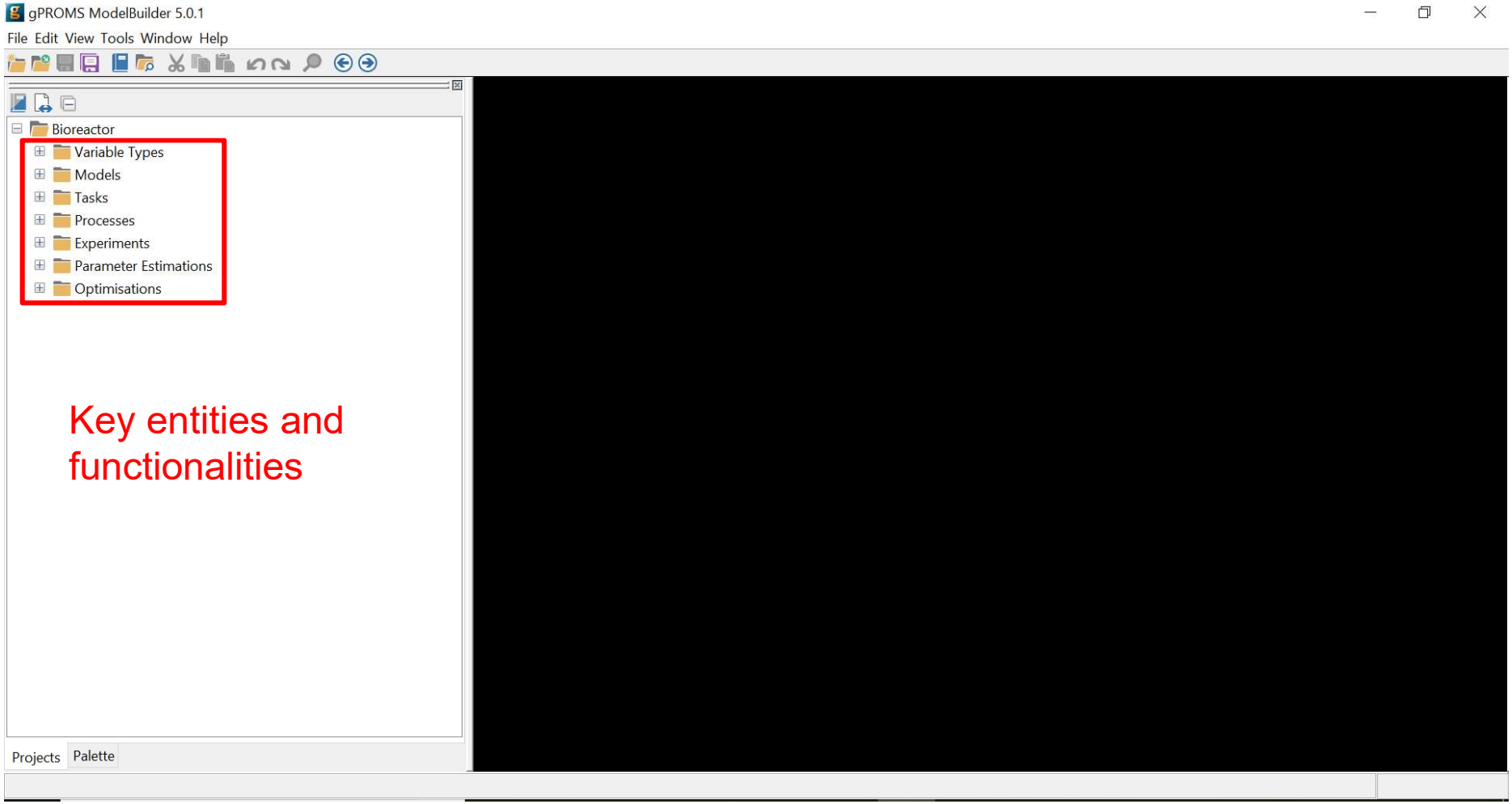
Dynamic process modelling

Seamless modelling capabilities (no manual discretization required)

Other functionalities: dynamic optimisation, parameter estimation based on experimental data, experiments to be designed

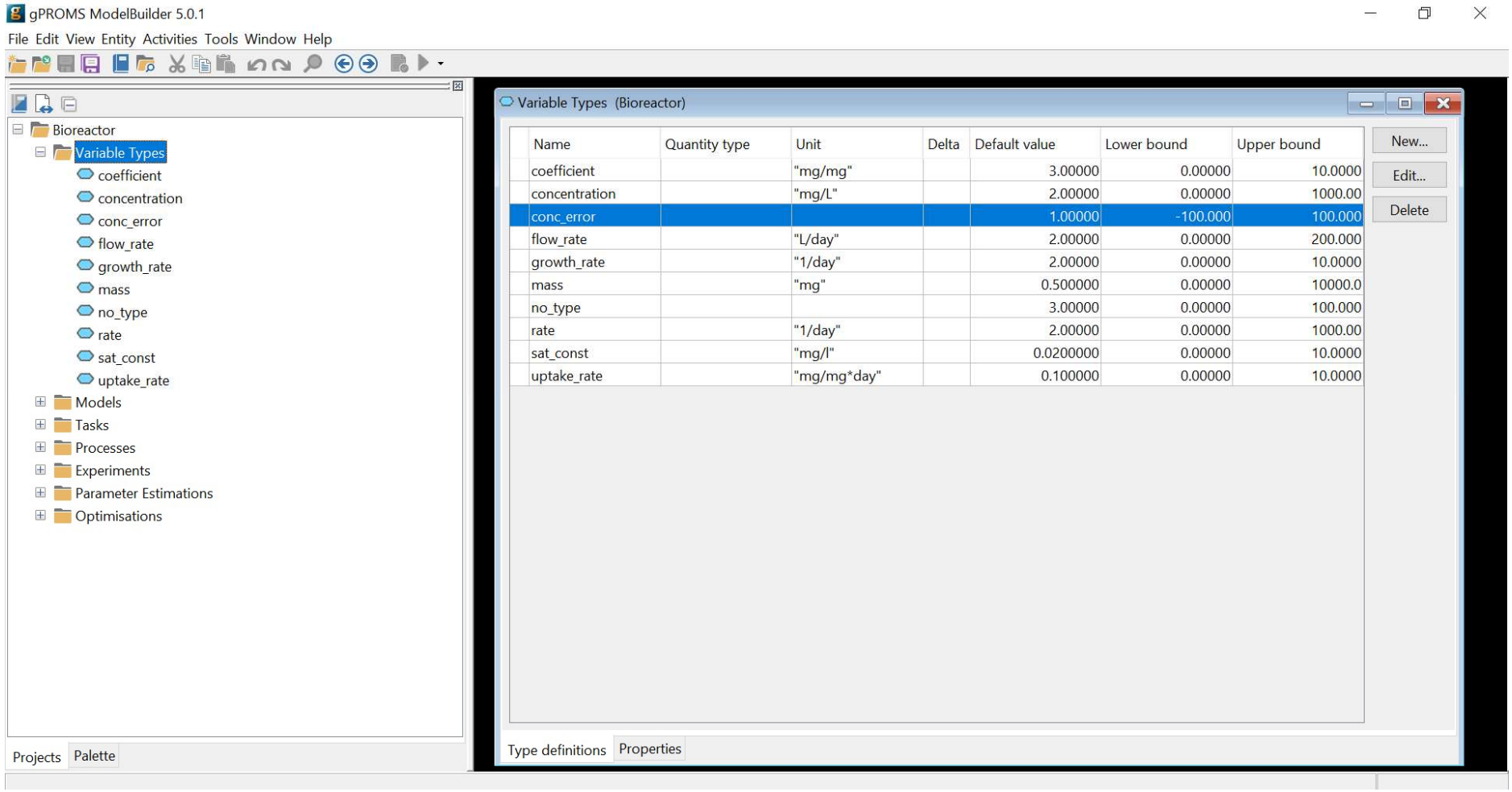
Imperial College London

Main navigation window



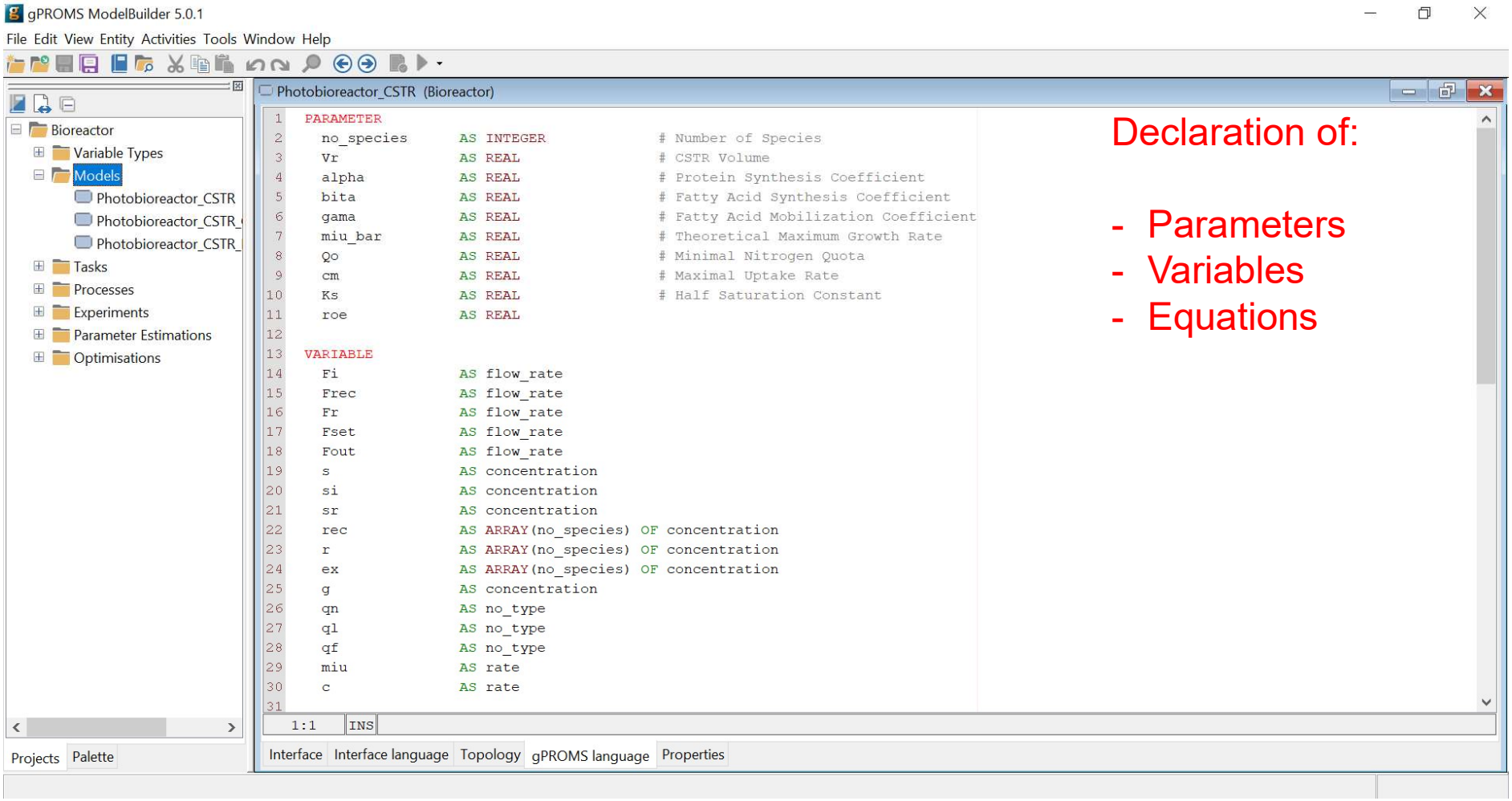
Imperial College London

Declaration of variables & bounds



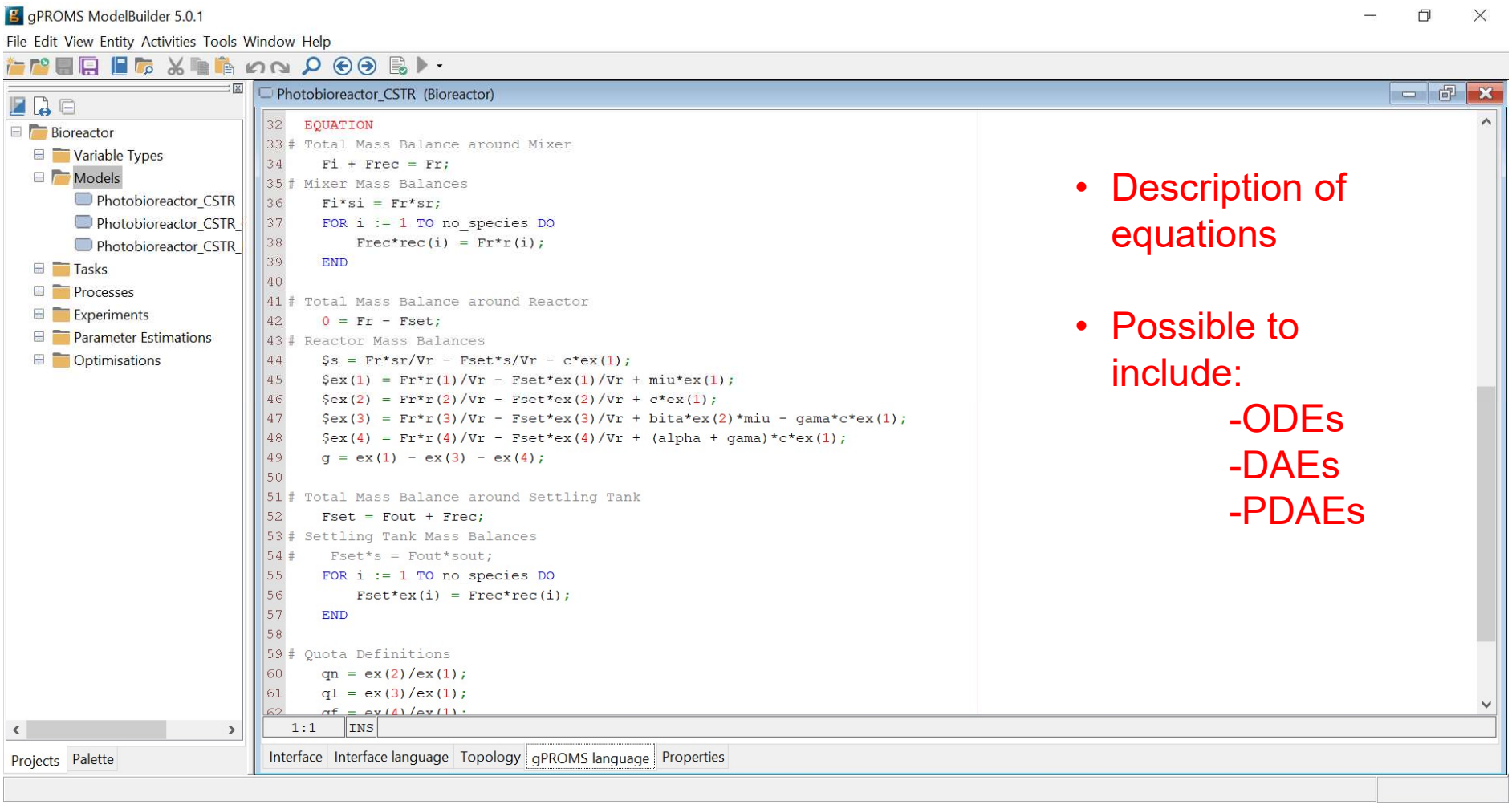
Imperial College London

Model entity



Imperial College London

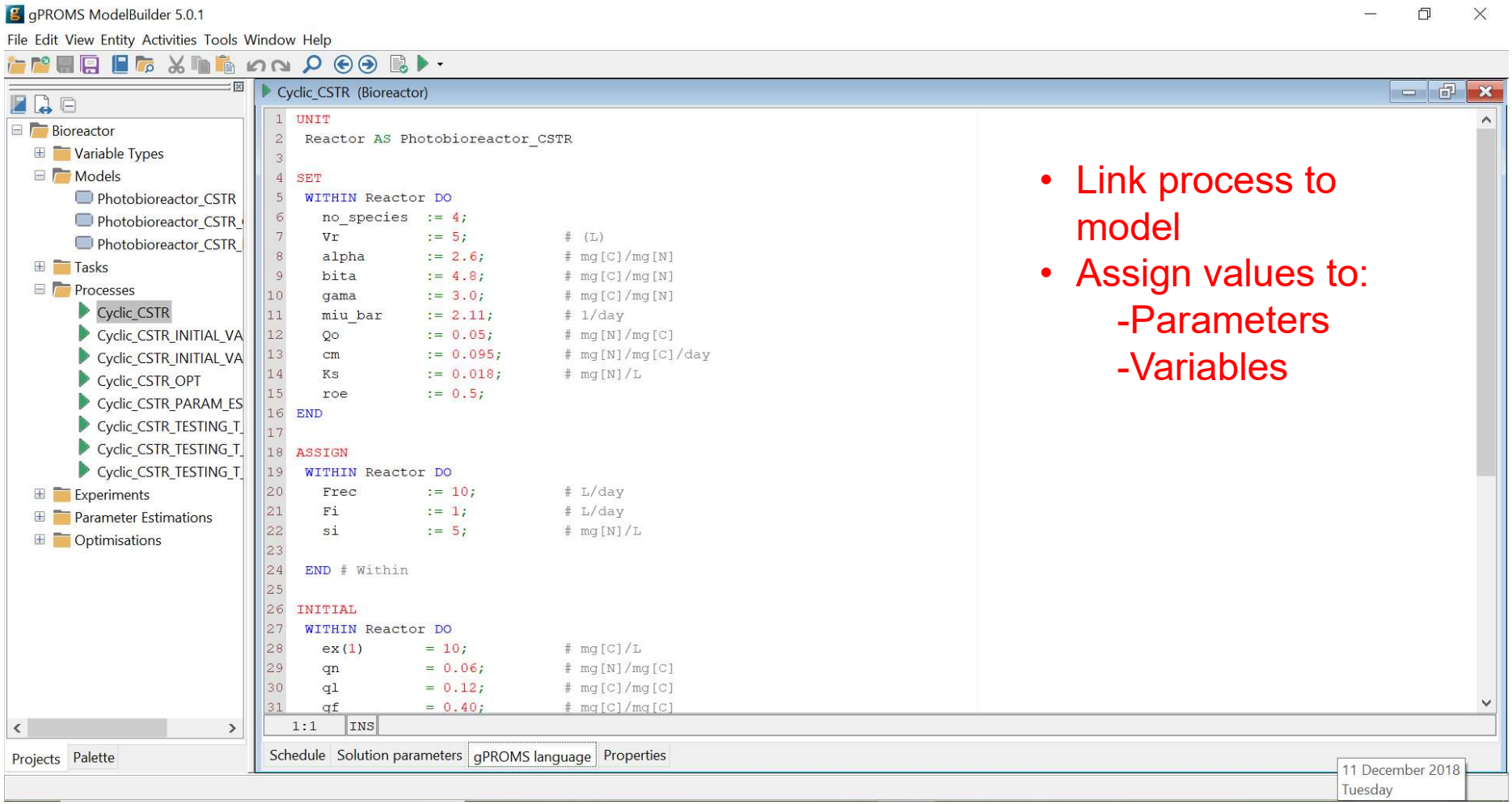
Model entity



- Description of equations
- Possible to include:
  - -ODEs
  - -DAEs
  - -PDAEs

Imperial College London

Process entity

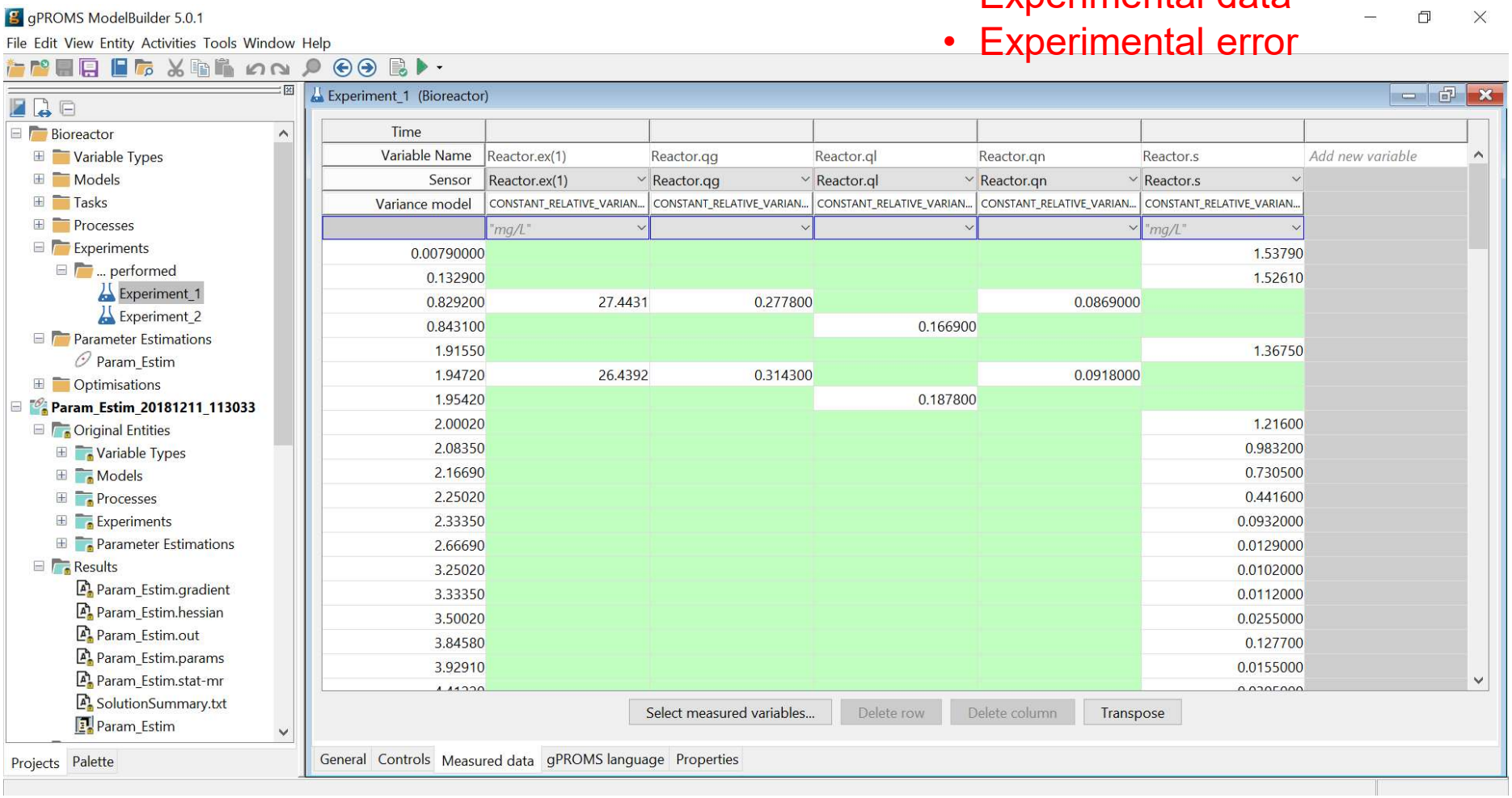


- Link process to model
- Assign values to:
  - -Parameters
  - -Variables

Imperial College London

Parameter estimation

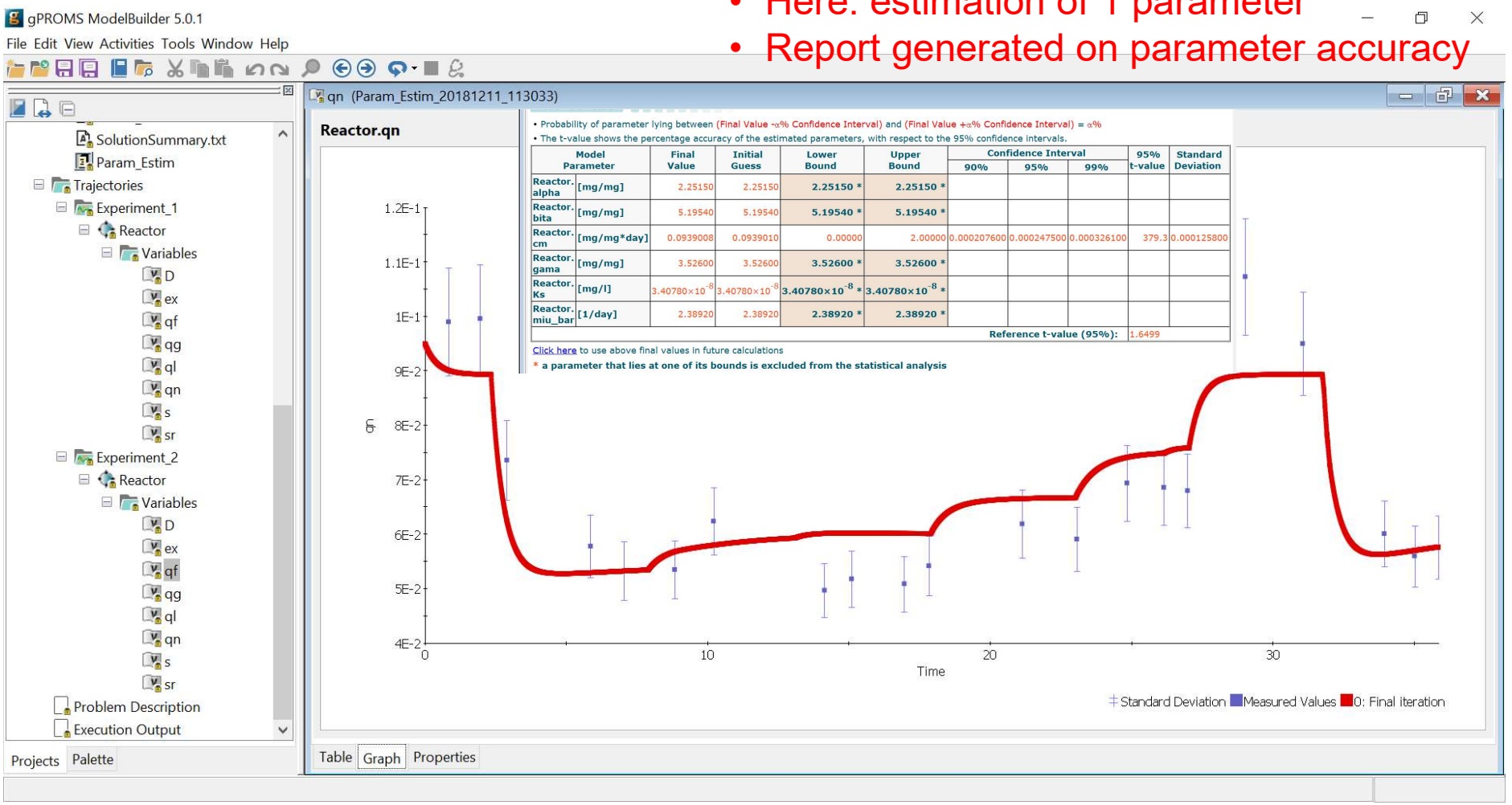
- Performed experiments
- Experimental data
- Experimental error



Imperial College London

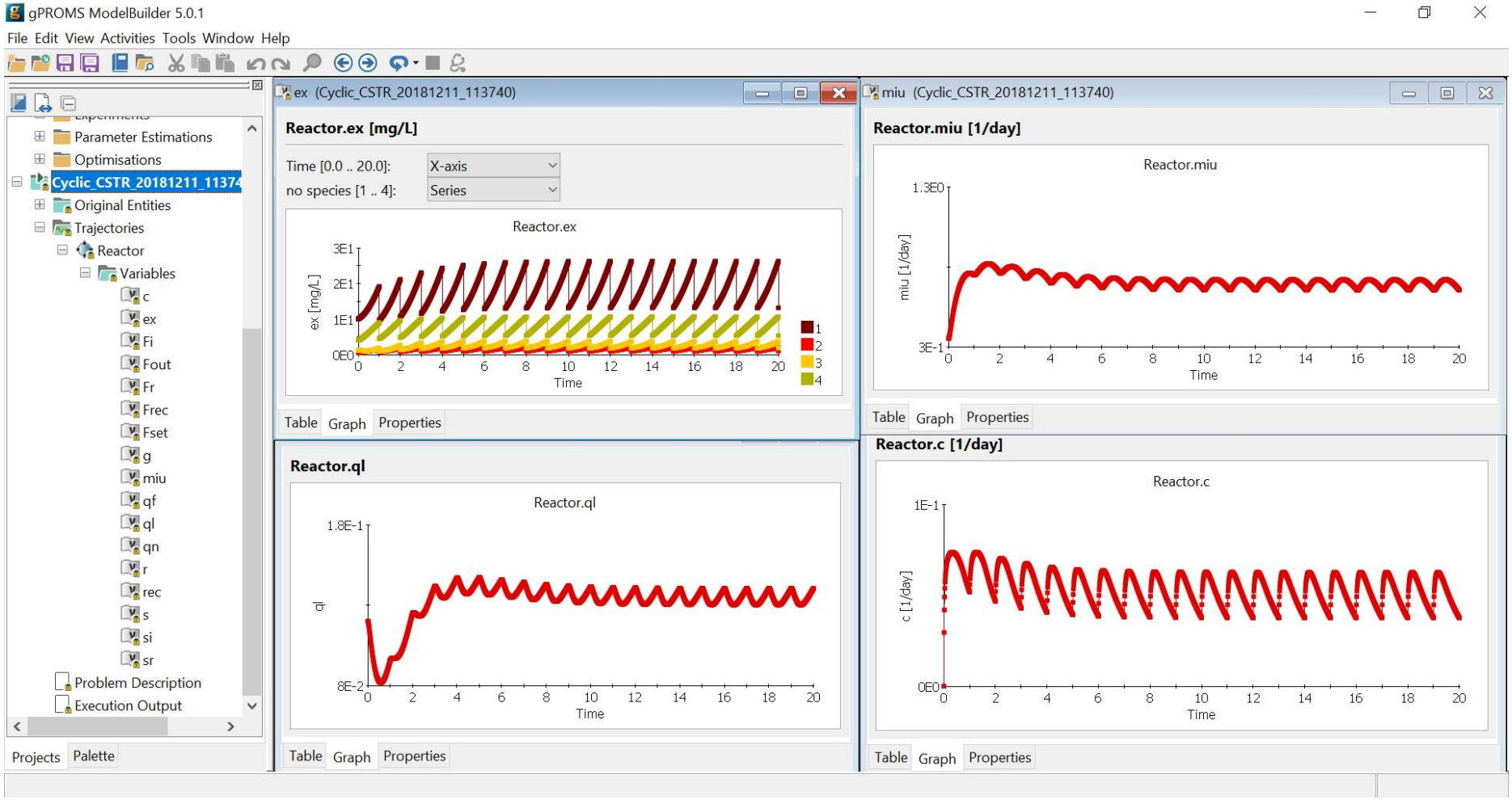
Parameter estimation

- Simulation vs experimental data
- Here: estimation of 1 parameter
- Report generated on parameter accuracy



Imperial College London

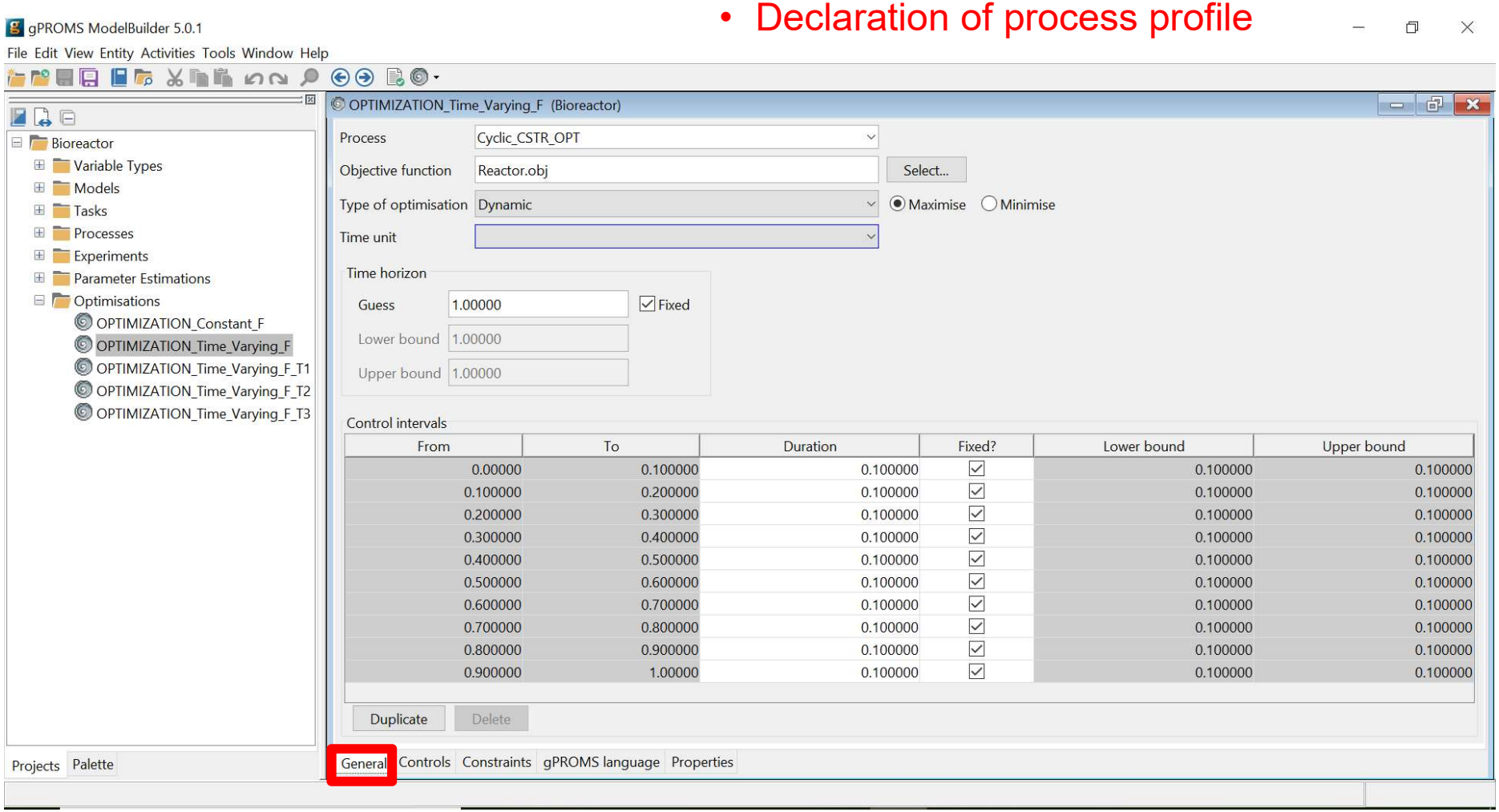
Open loop model simulation



Imperial College London

Dynamic optimisation

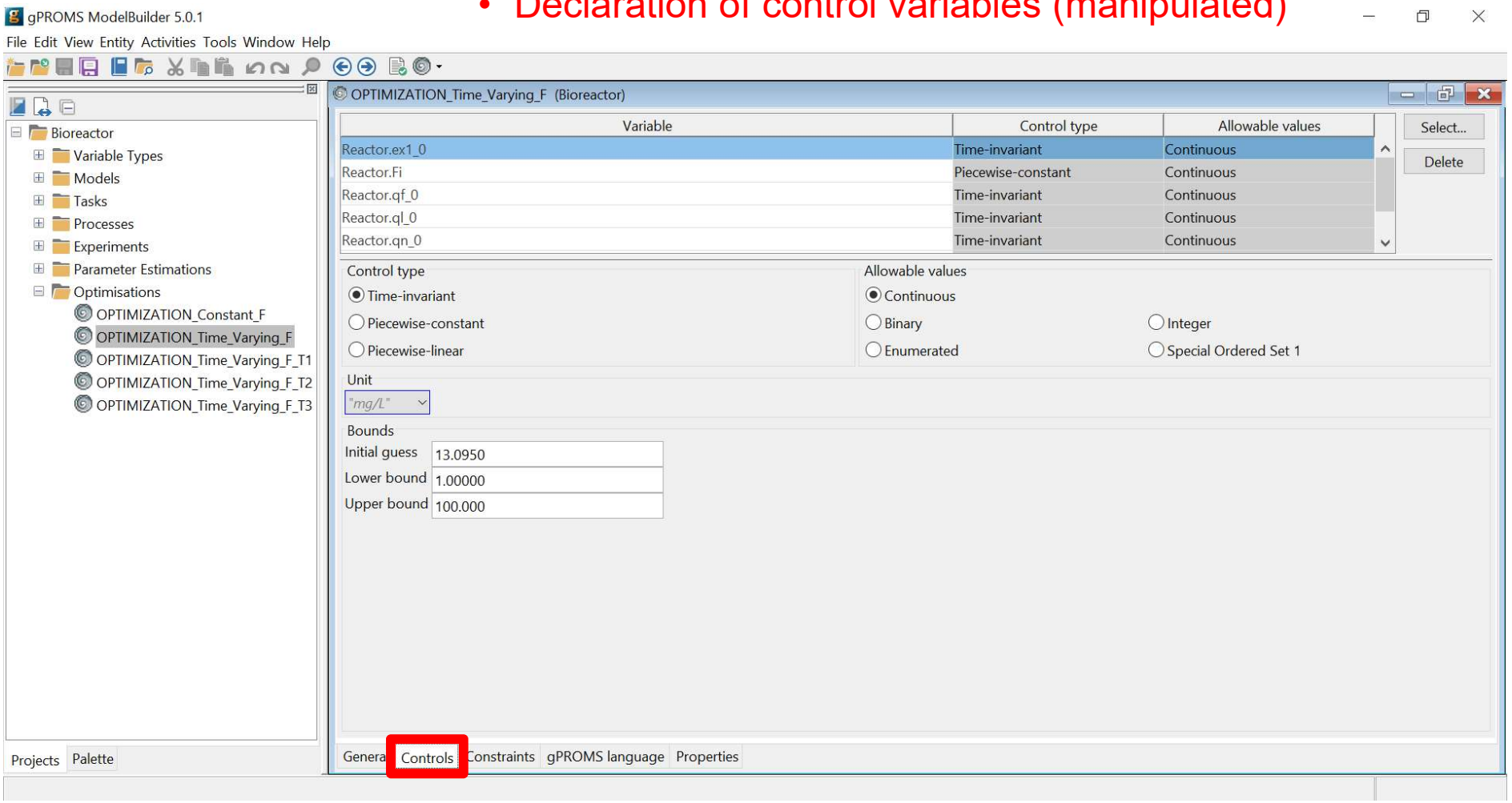
- Declaration of process profile



Imperial College London

Dynamic optimisation

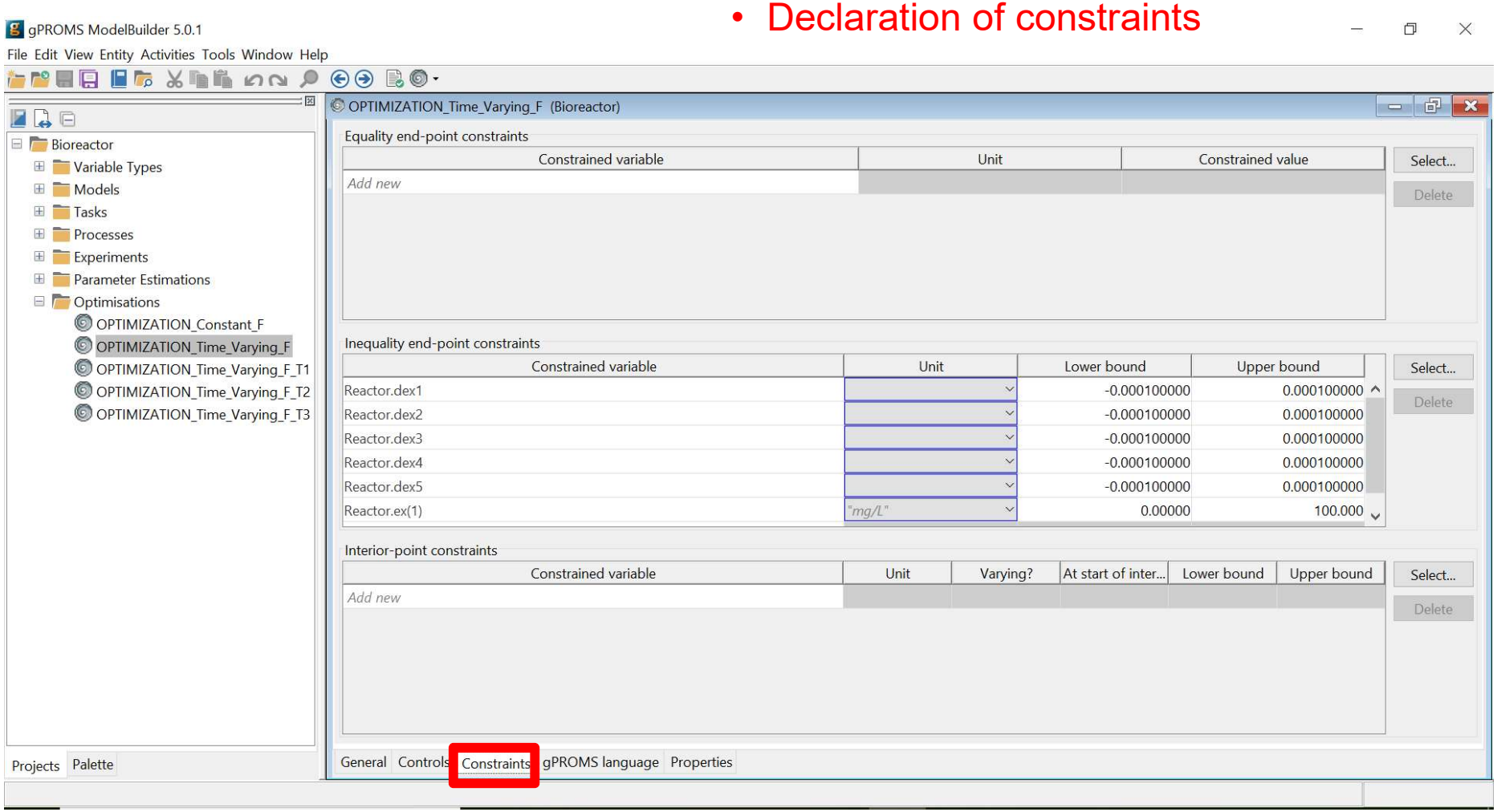
- Declaration of control variables (manipulated)



Imperial College London

Dynamic optimisation

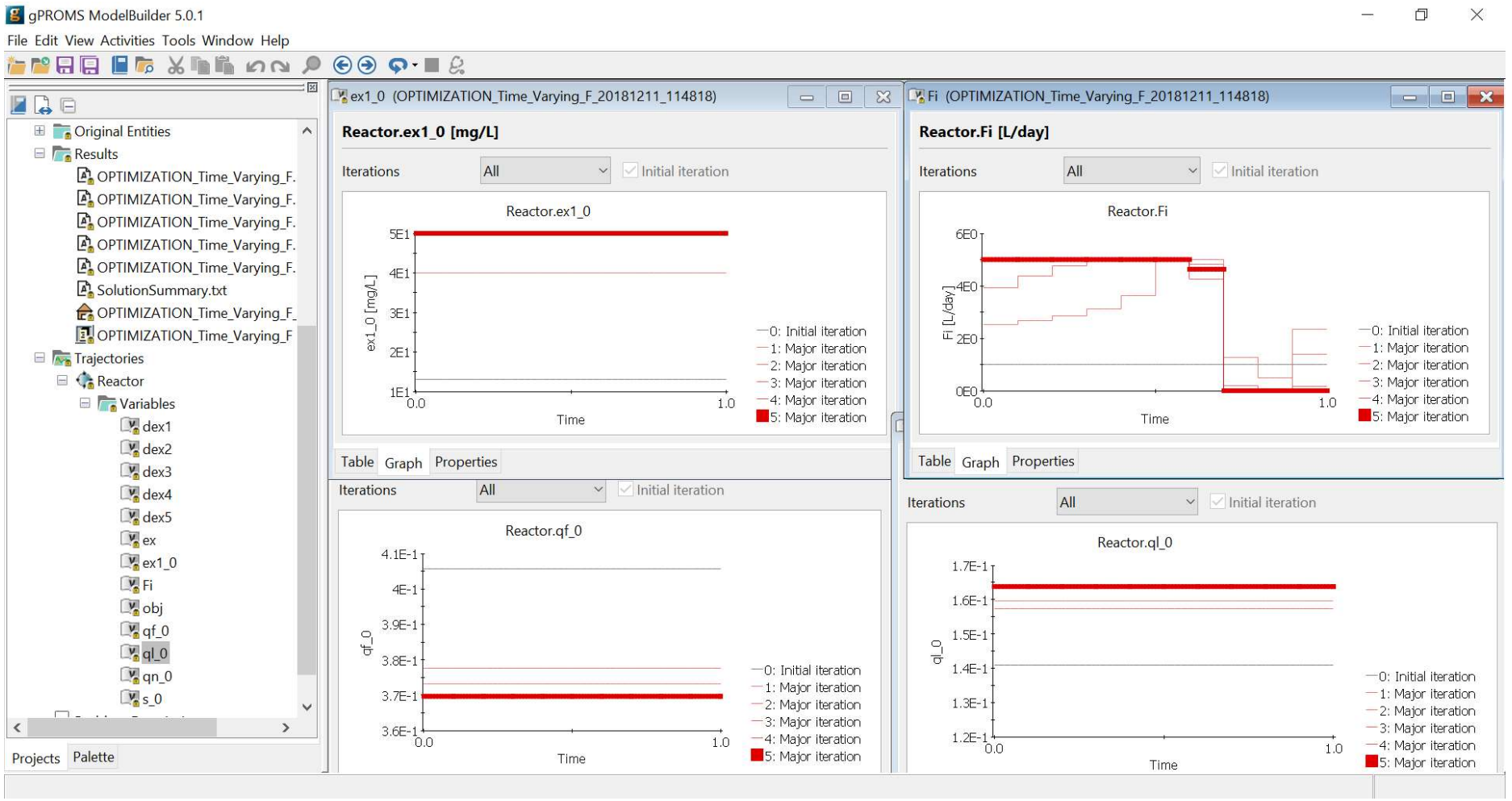
- Declaration of constraints



Imperial College London

Dynamic optimisation – Results

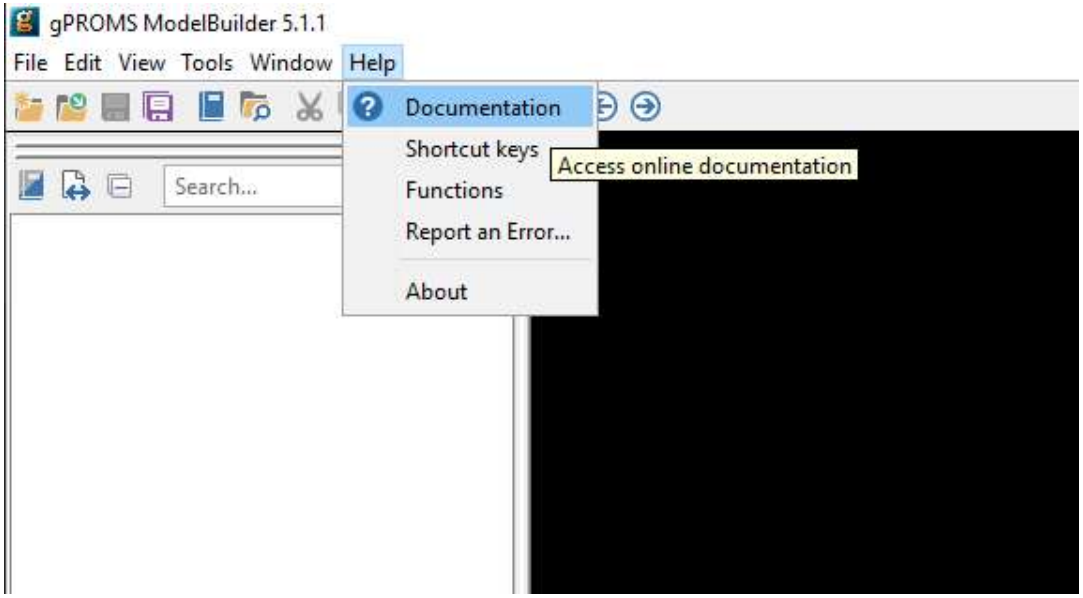
- Suggested optimal values following 5 iterations



Imperial College London

Getting Help

- In gPROMS click on “Help” menu then click on “Documentation”.



- PSE Webinars:
https://www.psenterprise.com/events/webinars

Imperial College London

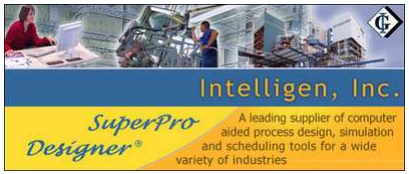
Outline

- Work Stream 1 (WS1): Improving existing processes and platforms ✓
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies ✓
- Modelling tools:
  - gPROMS (unit operation level); ✓
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).

Imperial College London

Introduction to SuperPro Designer

- Process flow sheet level modelling;
- Facilitates modelling, evaluation and optimization;
- Graphical user interface;
- Works by solving material and energy balances;
- Can be interfaced with Visual Basic for Applications (VBA), MatLab, Python, for automation and sensitivity analysis.

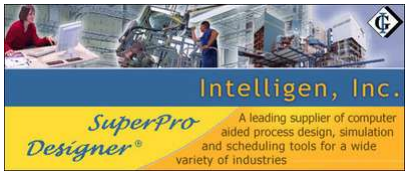


INTELLIGEN, INC.
Simulation, Design, and Scheduling Tools for the Process Manufacturing Industries

Imperial College London

SuperPro Designer – Key features

- Contains models for over 140 unit procedures/operations;
- Extensive chemical component and mixture database;
- Extensive equipment and resource databases;
- Equipment sizing and costing;
- Thorough process economics;
- Scheduling of batch operations;
- Throughput Analysis and Debottlenecking.

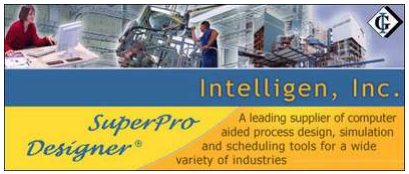


INTELLIGEN, INC.
Simulation, Design, and Scheduling Tools for the Process Manufacturing Industries

Imperial College London

SuperPro Designer – Typical questions being addressed

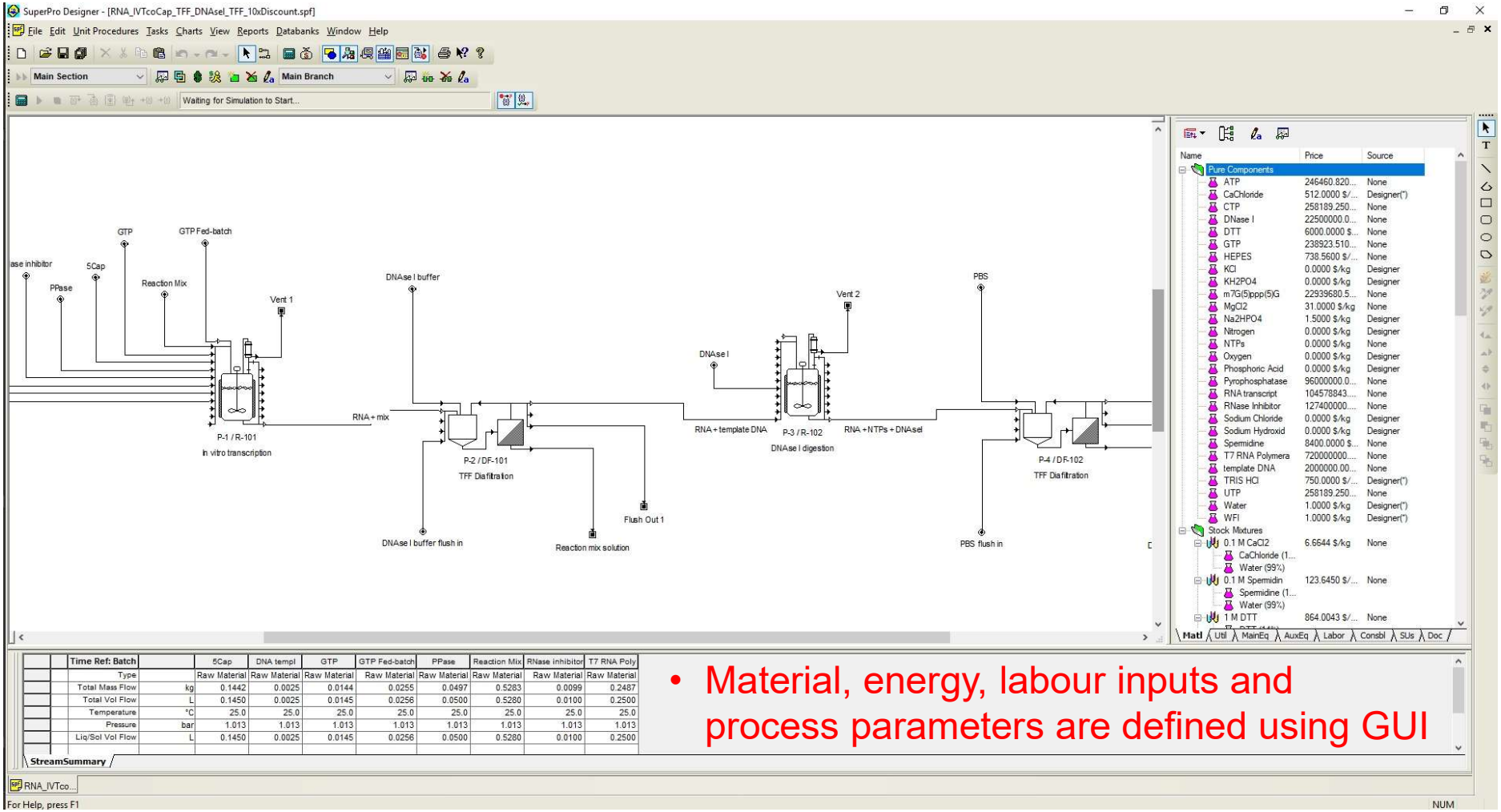
- How would the cost per dose change with increasing titres/yields?
- Which process design is more cost effective (by comparing different process designs)?
- How would the cost per dose and the upfront capital cost change at different production scales?
- Where are the production bottlenecks?
- Which are the major cost components and how to reduce those?
- How feasible is a continuous process compared to a batch process?



INTELLIGEN, INC.
Simulation, Design, and Scheduling Tools for the Process Manufacturing Industries

Imperial College London

SuperPro Designer – Graphical User Interface (GUI)



- Material, energy, labour inputs and process parameters are defined using GUI

Imperial College London

SuperPro Designer – Model inputs

- Type of unit procedures and parameters for each operation within procedures;
- Material inputs (default values are provided for labour and utilities);
- Definition of the reaction/fermentation (e.g. stoichiometric, kinetic, equilibrium) and reaction components;
- Specification of material flows between unit procedures;
- Specification of the sequence and duration of unit procedures (i.e. scheduling);
- Optional: a variety of other inputs can be specified (e.g. costs of equipment, costs of consumables, costs of labour and utilities, etc.)

Imperial College London

SuperPro Designer – Outputs

Charts:

- Equipment occupancy charts
- Operations Gantt Charts
- Equipment Gantt Charts
- Material, labour and utility utilization charts

Reports (MS Word, MS Excel or PDF formats):

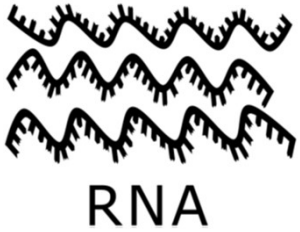
- Materials & Streams
- Economic Evaluation
- Cash Flow Analysis
- Itemized Cost
- Environmental Impact, etc.

Imperial College
London

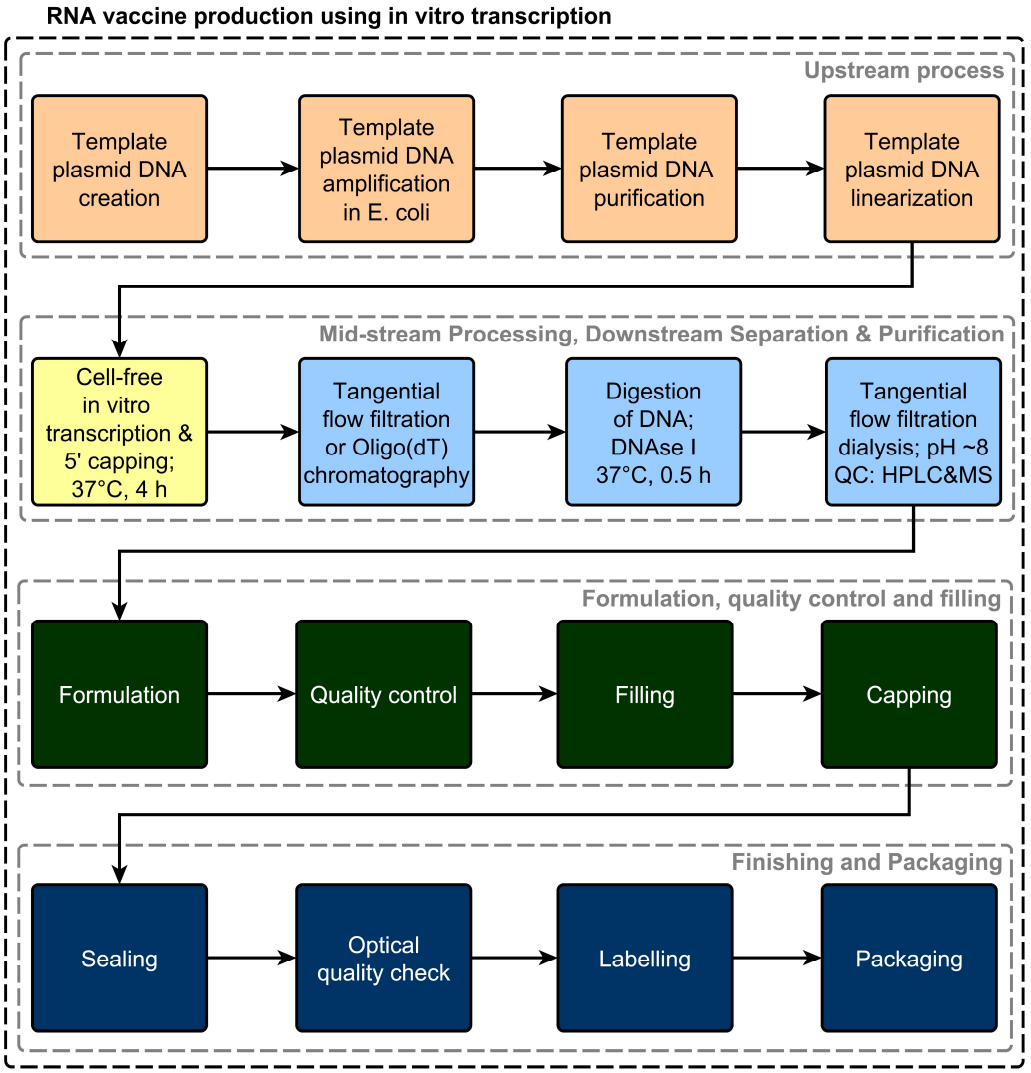
SuperPro Designer
Example Problem 1
“Cost modelling for new vaccine processes”

Imperial College London

RNA vaccines – how expensive? how to reduce costs?



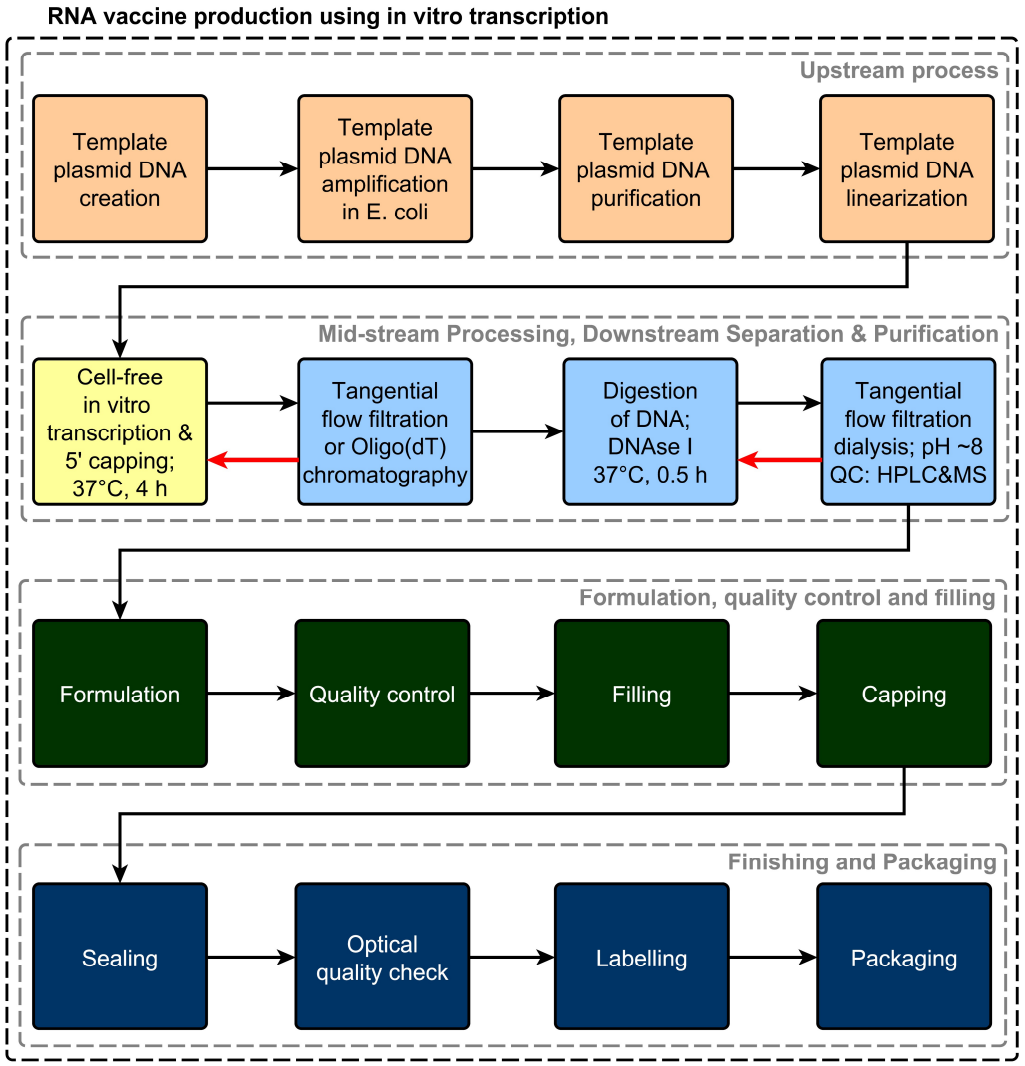
- ~10 kb self-amplifying RNA
- produced by *in vitro* transcription
- cell-free product
- Co-transcriptional 5' capping, ARCA



Imperial College London

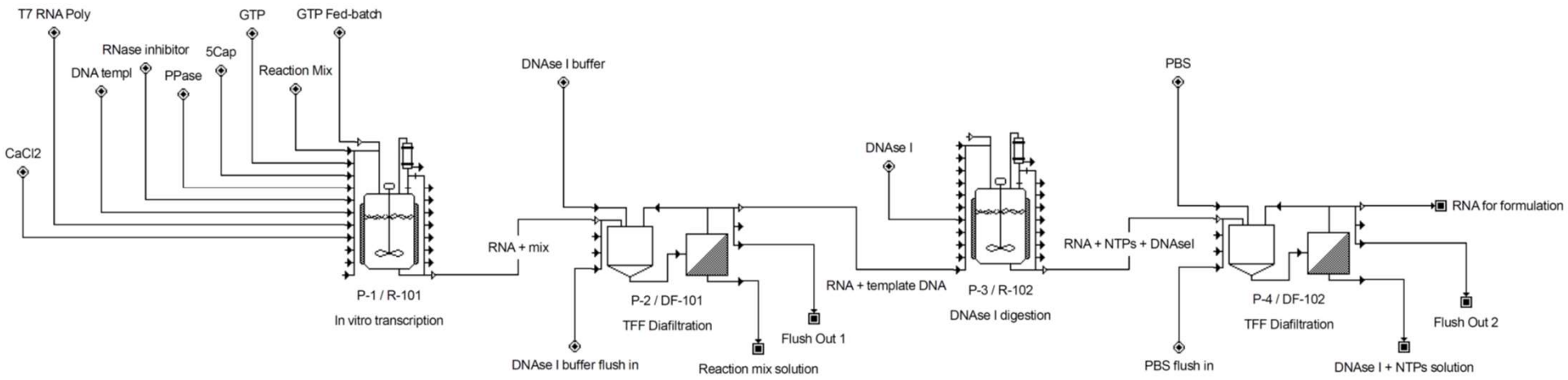
RNA vaccines – how expensive? how to reduce costs?

- ~10 kb self-amplifying RNA
- produced by *in vitro* transcription
- Co-transcriptional 5' capping, ARCA
- 5x Recycling of materials



Imperial College London

RNA production model in SuperPro Designer



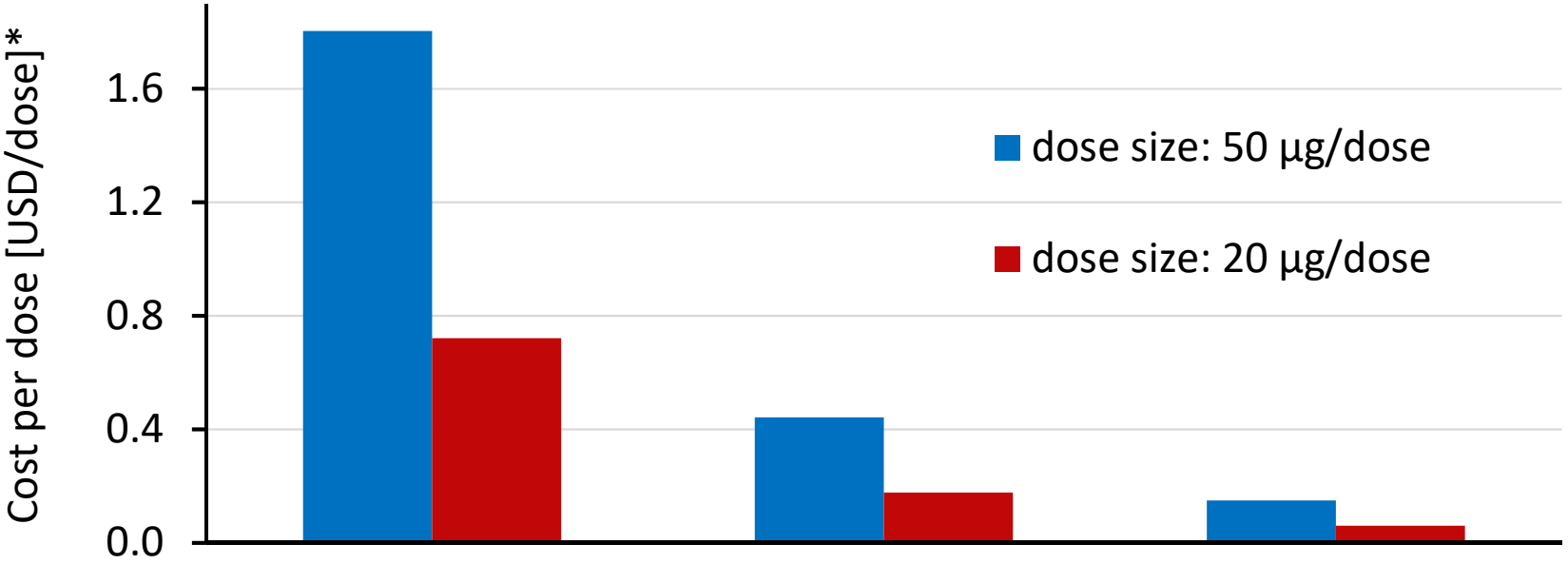
- ~10 kb self-amplifying RNA
- produced by *in vitro* transcription
- cell-free product
- Co-transcriptional 5' capping, ARCA

Cost per dose at 20 µg/dose at 1 L reaction volume: 0.72 USD/dose *

* Costs do not include formulation and secondary manufacturing (fill and finish) costs, upstream costs were accounted for indirectly. Default capital, labour, consumables, utilities, facility-related, maintenance and QA/QC costing values and methods from SuperPro Designer were used.

Imperial College London

RNA Cost modelling results

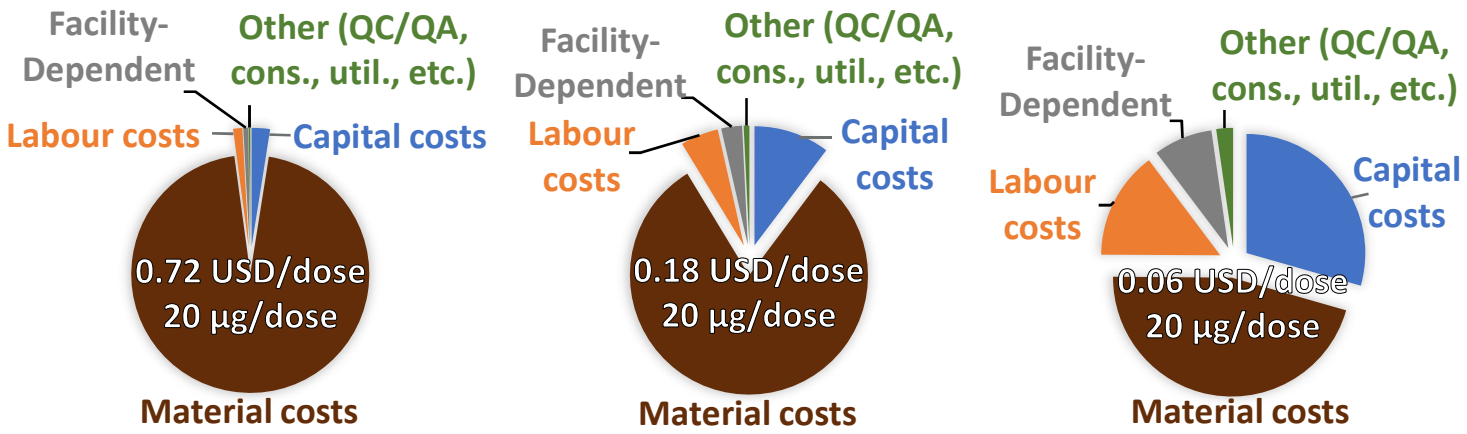


Recirculation?	No	5x	5x
5’ Cap type?	ARCA	ARCA	Non-capped

* Costs do not include formulation and secondary manufacturing costs, upstream costs accounted indirectly; Simulation results with default costing values and method from SuperPro Designer, capital costs were spread over 5 years.

Imperial College London

RNA Cost modelling results

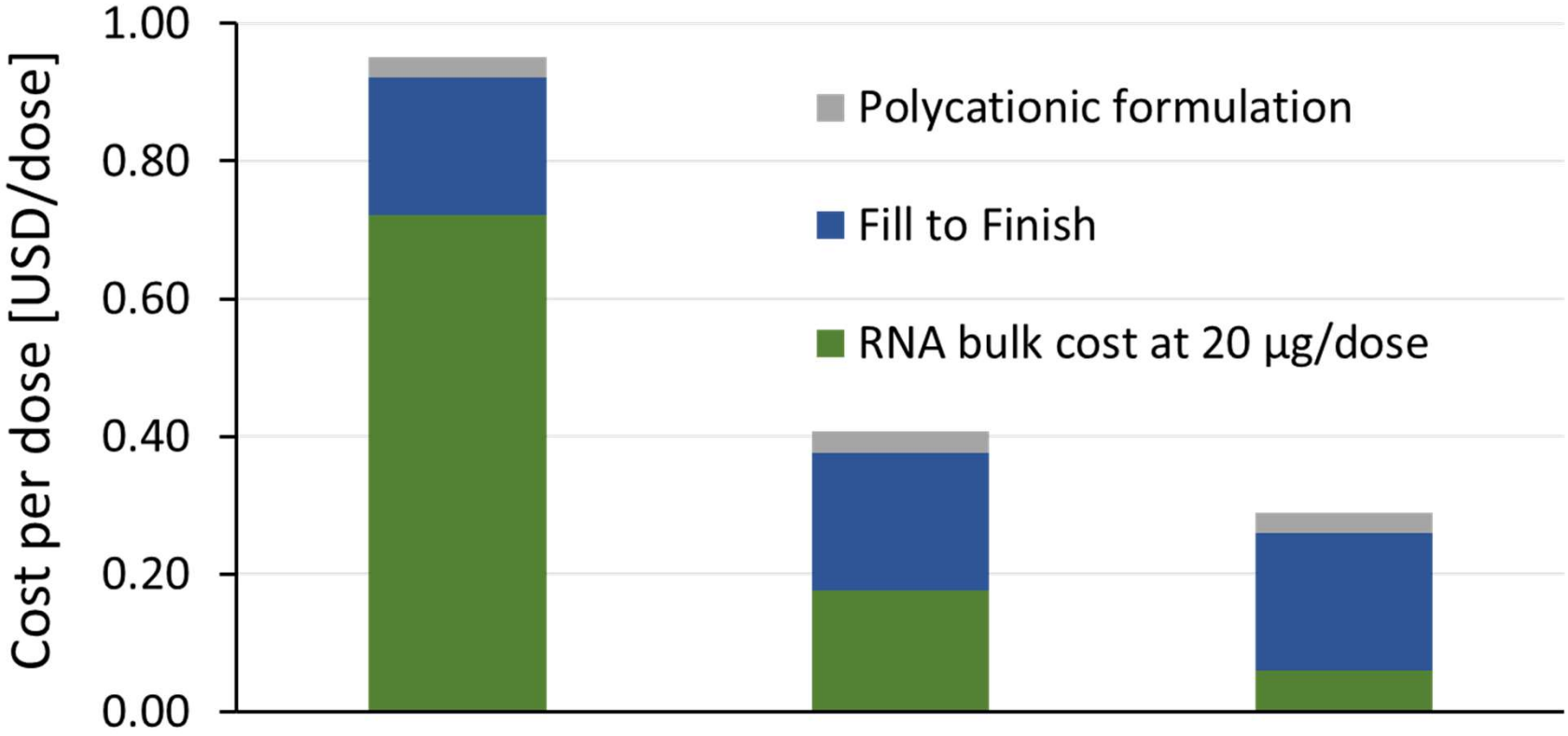


Recirculation?	No	5x	5x
5' Cap type?	ARCA	ARCA	Non-capped

* Costs do not include formulation and secondary manufacturing costs, upstream costs accounted indirectly;
Simulation results with default costing values and method from SuperPro Designer, capital costs were spread over 5 years.

Imperial College London

RNA Cost modelling results



Recirculation?	No	5x	5x
5’ Cap type?	ARCA	ARCA	Non-capped

Imperial College
London

Getting Help

- Pressing the F1 button;

- SuperPro Designer manual:
http://www.intelligen.com/downloads/SuperPro_ManualForPrinting_v10.pdf

- Training videos, from the Intelligen, Inc. website:
http://www.intelligen.com/videos.html

- Papers and literature, from the Intelligen, Inc. website:
http://www.intelligen.com/literature.html

- Attending training course offered by Intelligen, Inc.:
http://www.intelligen.com/training.html

Imperial College London

Outline

- Work Stream 1 (WS1): Improving existing processes and platforms ✓
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies ✓
- Modelling tools:
  - gPROMS (unit operation level); ✓
  - SuperPro Designer (process flow sheet level); ✓
  - GAMS (manufacturing and supply chain level).

Imperial College London

Outline

- Introduction to GAMS
- GAMS Example Problem
- Exercise

Latest GAMS version (demo) free download link:

http://www.gams.com/download/

Reference: “GAMS-A User's Guide”

Introduction to GAMS

General Algebraic Modelling System

is an

Algebraic Modelling Language (AML)

(high-level computer programming language)

main advantages

The **syntax** is **similar** to the **mathematical notation** of optimization problems.

The **programming** of the optimization model is **independent** of the **solution algorithm**.

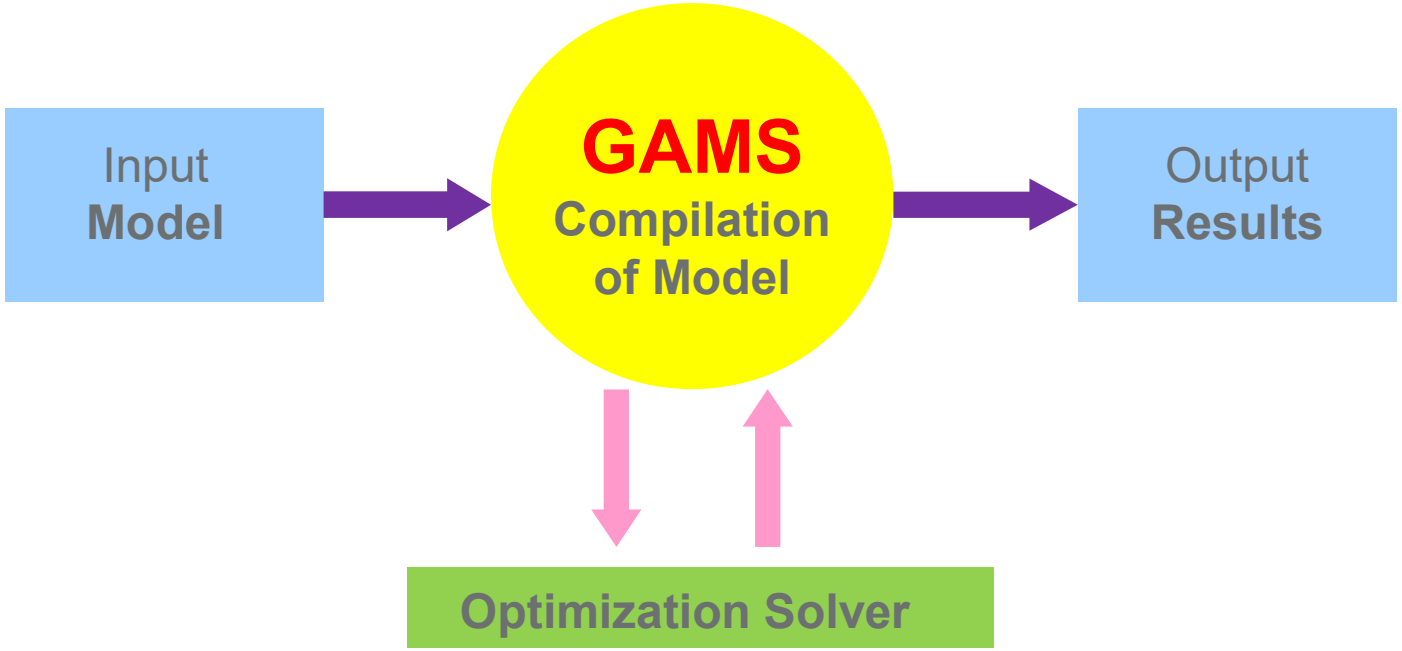
Imperial College London

How GAMS works

GAMS is a modeling system for optimization that **provides an interface with a variety of optimization algorithms** (solvers).

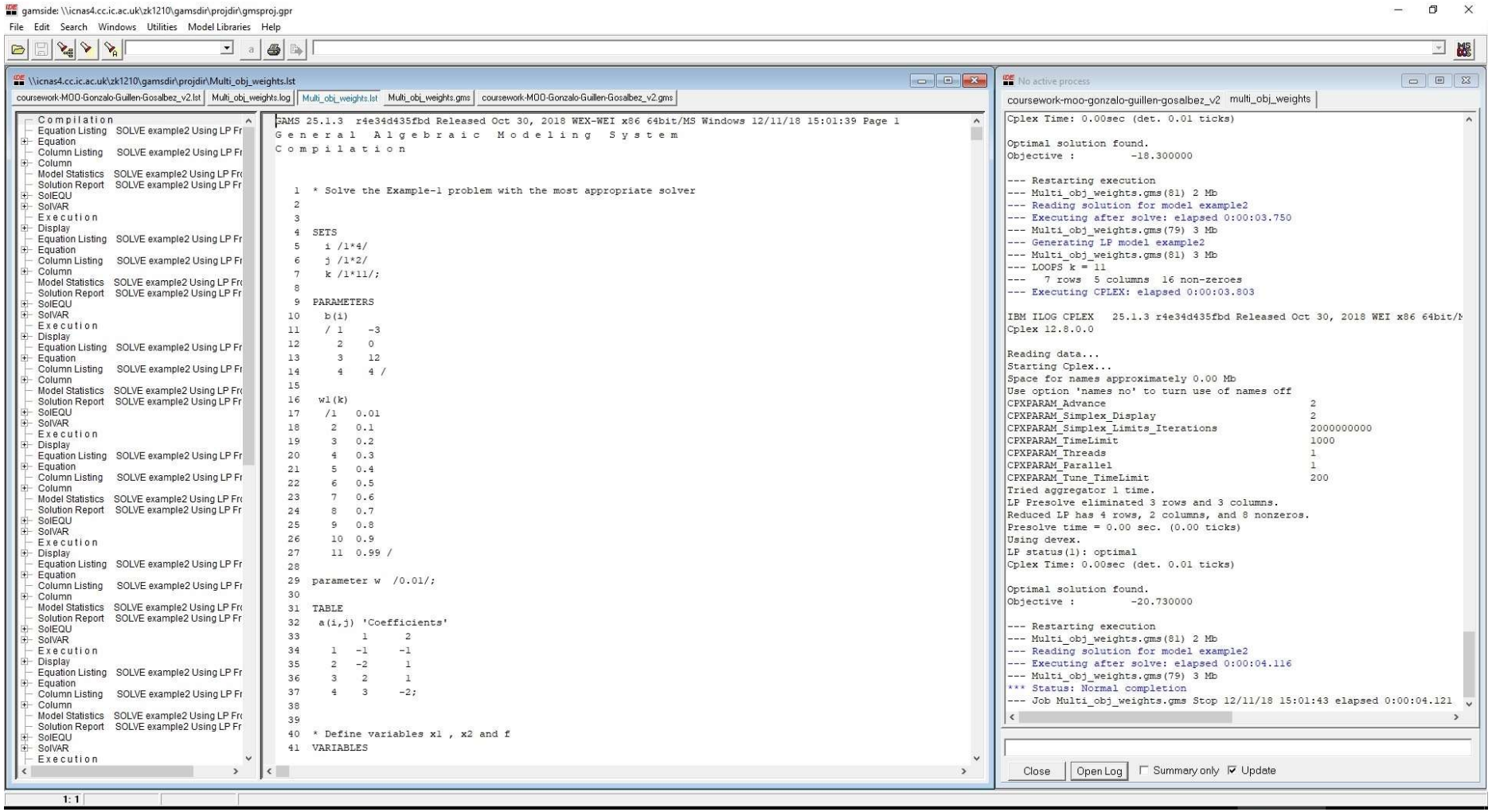
Users introduce the model to GAMS in the form of algebraic equations.

GAMS compiles the model and interfaces automatically with a solver.



Imperial College
London

GAMS user interface



Imperial College London

GAMS optimization model types and solvers

Main optimization model types in GAMS:

- **LP** Linear Programming
- **NLP** NonLinear Programming
- **MIP** Mixed Integer linear Programming
- **RMIP** relaxed MILP where the integer variables are treated as continuous
- **MINLP** Mixed Integer NonLinear Programming; involve integer variables and nonlinear equations
- **RMINLP** Relaxed MINLP where the integer variables are treated as continuous

Main solvers for each type of optimization model:

- **LP** **CPLEX**
- **MIP, RMIP** **CPLEX**
- **NLP** **BARON, MINOS, CONOPT, ANTIGONE**
- **MINLP** **BARON, DICOPT, ANTIGONE**

Imperial College London

Structure of a GAMS Model

SETS

- Declaration
- Assignment of members

PARAMETERS, TABLES, SCALARS (DATA)

- Declaration
- Assignment of values

VARIABLES

- Declaration
- Assignment of type
- Assignment of bounds and/or initial values (optional)

EQUATIONS

- Declaration
- Definition

MODEL and SOLVE statements

OPTION for output file and solver

DISPLAY statements (optional)

Imperial College London

Variables

- Types of variables:

Variable Type	Allowed Range of Variables
FREE	-∞ to +∞
POSITIVE	0 to +∞
NEGATIVE	-∞ to 0
BINARY	0 or 1
INTEGER	0, 1, …, 100

- The default type is `FREE`
- Remark: The variable being optimized must be `FREE` and not indexed

Imperial College London

Equations

The keyword **`EQUATIONS`** is for **listing the names** (which are random) of the constraints and objective function.

The equations are **defined** by: equation_name..

Syntax for the equality and inequality signs:

`=E=`	for =
`=G=`	for ≥
`=L=`	for ≤

The basic arithmetic operators are:

`+`	addition
`-`	subtraction
`*`	multiplication
`/`	division
`**`	exponent

*The **semi-colon ;** is needed at the end of each equation.*

Example: equation_name.. variable_1 + variable_2 =E= 1;

Imperial College London

Bounds and Initial Values

Bounds and initial values can be provided by adding a **suffix** to the variables.

Syntax for specifying bounds and initial values:

`(variable name).LO` lower bound (e.g., `x.LO`)
`(variable name).UP` upper bound (e.g., `x.UP`)
`(variable name).FX` fixed value
`(variable name).L` level value, meaning actual value (initial or final)
`(variable name).M` dual prices, Lagrange or Kuhn-Tucker multipliers

No need to specify lower bounds of zero for variables defined as **`POSITIVE VARIABLE`**.

In general, it is not a requirement to specify initial values for the variables.

However, for **nonlinear** models it is often advisable to provide an **initial guess** (e.g., `X.L=4;`).

Imperial College London

Keywords

MODEL

The keyword **MODEL** is used to **name the optimization model** and to **specify which equations are included in it**.

```
MODEL (model name) /ALL/;
          OR
MODEL (model name) /eq1,eq2/;
```

OPTION

The keyword **OPTION** is used to **suppress output** for debugging the compilation of the equations, and to **set options for solvers** (max CPU s).

SOLVE

The keyword **SOLVE** **calls the optimization solver**. The syntax is as follows:

```
SOLVE (model name) USING (solver type) MINIMIZING (objective variable);
```

DISPLAY

The keyword **DISPLAY** shows the values of the requested symbols.

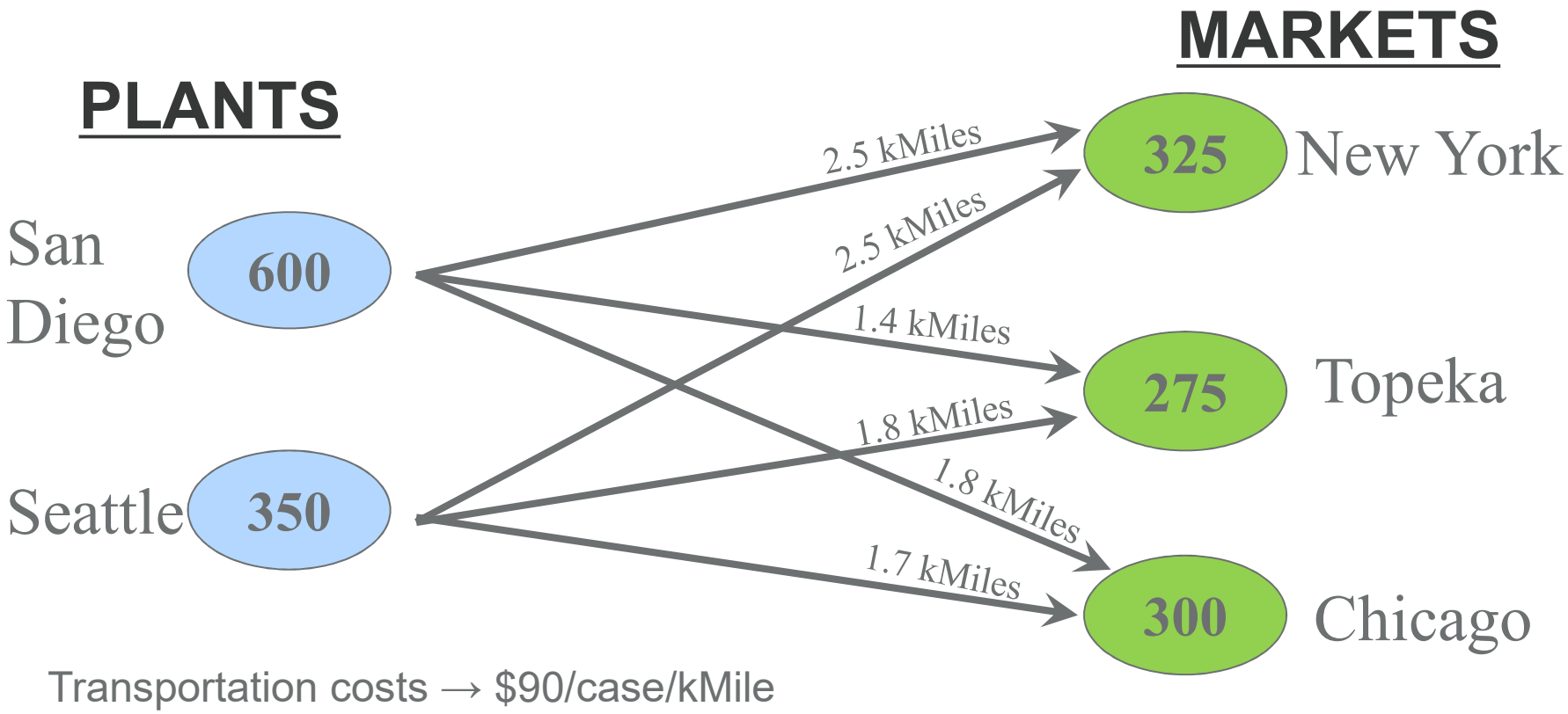
```
DISPLAY (variable name).suffix;   and   DISPLAY (parameter or set name);
```

Imperial College
London

GAMS
Example Problem 1
Transportation problem

Imperial College London

A Transportation Problem



Transportation costs → $90/case/kMile

Objective → **minimize transportation cost**

Subject to: **demand satisfaction & supply constraints**

Imperial College London

A Transportation Problem

Indices (or sets):

i : plants (San Diego, Seattle)

j : markets (New York, Topeka, Chicago)

Given Data (or parameters):

a_i : supply of commodity of plant i (in cases)

b_j : demand for commodity at market j (cases)

c_{ij} : transportation unit cost between plant i and market j ($/case)

Imperial College London

A Transportation Problem

Distances d_{ij} (in KMile)				
	Markets			
Plants	**New York**	**Chicago**	**Topeka**	*supply*
Seattle	2.5	1.7	1.8	350
San Diego	2.5	1.8	1.4	600
demand	325	300	275	

Transportation costs → \$90/case/kMile.

$$c_{ij} = \frac{90 \cdot d_{ij}}{1000}$$

Imperial College London

A Transportation Problem

Decision Variables:

x_{ij} : amount to transfer from plant i to market j. $x_{ij} \geq 0, \quad \forall i, j$

Objective Function:

Minimize $\sum_i \sum_j c_{ij} x_{ij}$ ($K)

Constraints:

Supply limit in plant i: $\sum_j x_{ij} \leq a_i, \quad \forall i$

Satisfy demand at market j: $\sum_i x_{ij} \leq b_j, \quad \forall j$

Imperial College London

Representation in GAMS (Input)

```
Sets    i   plants       / seattle, san-diego /
        j   markets      / new-york, chicago, topeka / ;

Parameters
        a(i)  capacity of plant i in cases  / seattle 350,san-diego 600 /
        b(j)  demand at market j in cases   /    new-york    325
                                                 chicago     300
                                                 topeka      275  / ;
Table   d(i,j)  distance in thousands of miles
                        new-york       chicago      topeka
      seattle           2.5            1.7          1.8
      san-diego         2.5            1.8          1.4  ;

Scalar      f     freight in dollars per case per thousand miles  /90/ ;
Parameter   c(i,j)  transport cost in thousands of dollars per case ;
            c(i,j) = f*d(i,j)/1000;

Variables   x(i,j)  shipment quantities in cases
            z       total transportation costs in thousands of dollars ;

Positive Variable x ;   z.lo = 1;

Equations   cost        define objective function
            supply(i)   observe supply limit at plant i
            demand(j)   satisfy demand at market j ;

cost ..        z  =e=  sum((i,j), c(i,j)*x(i,j)) ;

supply(i) ..   sum(j, x(i,j))  =l=  a(i) ;

demand(j) ..   sum(i, x(i,j))  =g=  b(j) ;

Model transport /all/ ;

Solve transport Using LP Minimizing z;
Display x.L, x.M;
```

Representation in GAMS (Input)

$$z = \sum_i \sum_j c_{ij} x_{ij}$$

$$\sum_j x_{ij} \le a_i, \quad \forall i$$

$$\sum_i x_{ij} \le b_j, \quad \forall j$$

Imperial College
London

General notes

- Declare sets and variables first! You cannot refer to something that has not been defined!
- Terminate statements with semi-colons (;)
- GAMS compiler is not case-sensitive
- Lines starting with * are comment lines
- Names must start with a letter
- Descriptive text must:
  - fit on one line, and be no more than 80 characters long
  - not start with GAMS' reserved words or contain the symbols =, ;/

Imperial College London

Sets

- **Sets in GAMS ≡ indices in algebraic models, e.g.,**

```
Sets     i   canning plants   / seattle, san-diego /
         j   markets          / new-york, chicago, topeka / ;
```

- **Multi-words not allowed**: NEW-YORK not NEW YORK

- Can also write as:

```
Set      i   canning plants   / seattle, san-diego /
Set      j   markets          / new-york, chicago, topeka / ;
```

- **Use of the asterisk in set assignment**:

```
Set      T  time periods   / 1991*2000 /
         M machines        / MACH1*MACH24 /;
```

This corresponds to T = {1991, 1992, …, 2000} and M = {MACH1, MACH2, …, MACH24}

- **ALIAS** (I, IP); defines the SET (index) IP identical to the SET I

Imperial College London

Input Data

- **Entry by lists:**

```
Parameters a(i)  capacity of plant i in cases
                /    seattle     350
                     san-diego   600  /
           b(j)  demand at market j in cases
                /    new-york    325
                     chicago     300
                     topeka      275  / ;
```

- **Entry by Tables:**

```
Table d(i,j)  distance in thousands of miles
                         new-york       chicago      topeka
        seattle            2.5            1.7          1.8
        san-diego          2.5            1.8          1.4  ;
```

Alternatively: `d("seattle","new-york")=2.5; etc.`

- **Entry by Direct Assignment:**

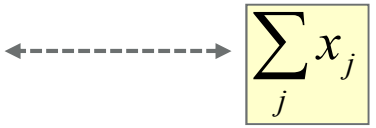
```
Parameter c(i,j)  transport cost in thousands of dollars per case ;
        c(i,j)=f*d(i,j)/1000;
```

- **Zero is default value for all un-assigned parameters and scalars**

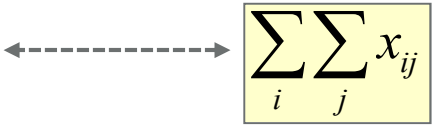
Imperial College London

Equations

- Note: equations must be declared and then defined
- **Use of SUM and PROD operators:**

`SUM(j,x(j))` 

`PROD(j,x(j))` 

`SUM((i,j),x(i,j))` *or* `SUM(i,SUM(j,x(i,j)))` 

Also, they can be used in direct assignment of PARAMETERS and SCALARS, e.g.,

```
SCALAR          TOTSUPPLY       total supply over all plants;
TOTSUPPLY = SUM(i, b(i));
```

Imperial College London

The Dollar Operator ($)

Provides a concise exception-handling capability

- **Example 1:** If $y \geq 1.5$, then $x = 2$, else $x = 1$

```
SCALAR X,Y;
Y = 2; X = 1;
X = 2$(Y GE 1.5);
```

- **Example 2:** If $y \leq 1.5$, then $x = 2$, else $x = 1$

```
SCALAR X,Y;
Y = 2; X = 1;
X$(Y LE 1.5) = 2;
```

- **Example 3:** If $x_i \neq 0$, then $\rho_i = 1/x_i$, else $\rho_i = 0$

```
rho(i) = (1/x(i))$(x(i) NE 0);
```

- **Example 4** (Equations): j = {1,2,3,4}. $\sum_{j=2}^{4} x_j = 1$

```
EQ1.. SUM(j$(j>1),x(j)) =E= 1;
```

- **Example 5** (Equations): If $x_i \neq 0$, then $z_i = y_i - 3$

```
EQ2$(x(i) NE 0).. z(i) =E= y(i) - 3;
```

Imperial College
London

GAMS Compilation

- Open **GAMS IDE**
- Create a **GAMS project file** `(project name).gpr`
- Create a **GAMS input file** `(file name).gms` and save it in the same directory with the project file
- Run GAMS by pressing **F9** or the **run button** in the GAMS IDE
- After the **compilation** of the **.gms** file, an output file is created `(file name).lst`

Imperial College London

The GAMS Output

The main elements of the GAMS output are:

- Echo Print
- Error Messages
- Equation Listings
- Model Statistics
- Status Reports
- Solution Reports

Imperial College London

Solution Reports

- The input statement **DISPLAY x.L, x.M;**


```
----     65 VARIABLE x.L  shipment quantities in cases

                new-york      chicago       topeka

seattle                       300.000
san-diego       325.000                     275.000


----     65 VARIABLE x.M  shipment quantities in cases

                new-york      chicago       topeka

seattle              EPS                        0.036
san-diego                        0.009
```

Imperial College
London

Final Remarks

Some other useful operators are the following:

- Dollar operator (**$**): it can be used to restrict the elements of a set; roughly speaking, it has the function of command IF.
- **LOOP** keyword which can be used to perform repetitive calculations for parameters or sets.
- **FOR** can be used for multiple **SOLVE** statements in an iterative algorithm.
- There are also other usual programming commands; e.g., **IF** and **WHILE**.

GAMS is available for several operating systems (Windows, Linux, Mac, Solaris).

GAMS is free, but the demo version could solve problems of limited size.

Solvers are NOT free (license is needed for every solver).

Other Algebraic Modeling Languages: *AMPL, AIMMS, etc.*

Imperial College London

Example Problem 2

A maximum-profit production problem

A furniture company wants to maximize its profits from the manufacture of four different types of desks.

Each different desk type requires different number of man-hours in each of the company sections, carpentry and finishing.

The profit per unit of every different type of desk sold is different and given.

The capacity of man-hours for carpentry and finishing is also given.

Labour requirements (man-hours)

	d1	d2	d3	d4
Carpentry	4	9	7	10
Finishing	1	1	3	40

Profit per unit sold

d1	d2	d3	d4
12	20	18	40

Capacity (man-hours)

Carpentry	6000
Finishing	4000

Formulate the optimization problem, and then introduce it to GAMS and solve it.

Imperial College
London

Problem Formulation

Sets: i: desk type (d1, d2, d3, d4)

j: sector (carpentry, finishing)

Data: caplim (j) : capacity limit of sector j

profit (i) : profit form selling desk type i

labor (j, i) : labor requirements (man-hours)

Decision Variable: x_i : number of desks i produced

Objective Function : $\sum_i x_i profit_i$

Constrain: $\sum_i labor_{j,i}\, x_i \leq caplim_j \; \forall j$

Imperial College
London

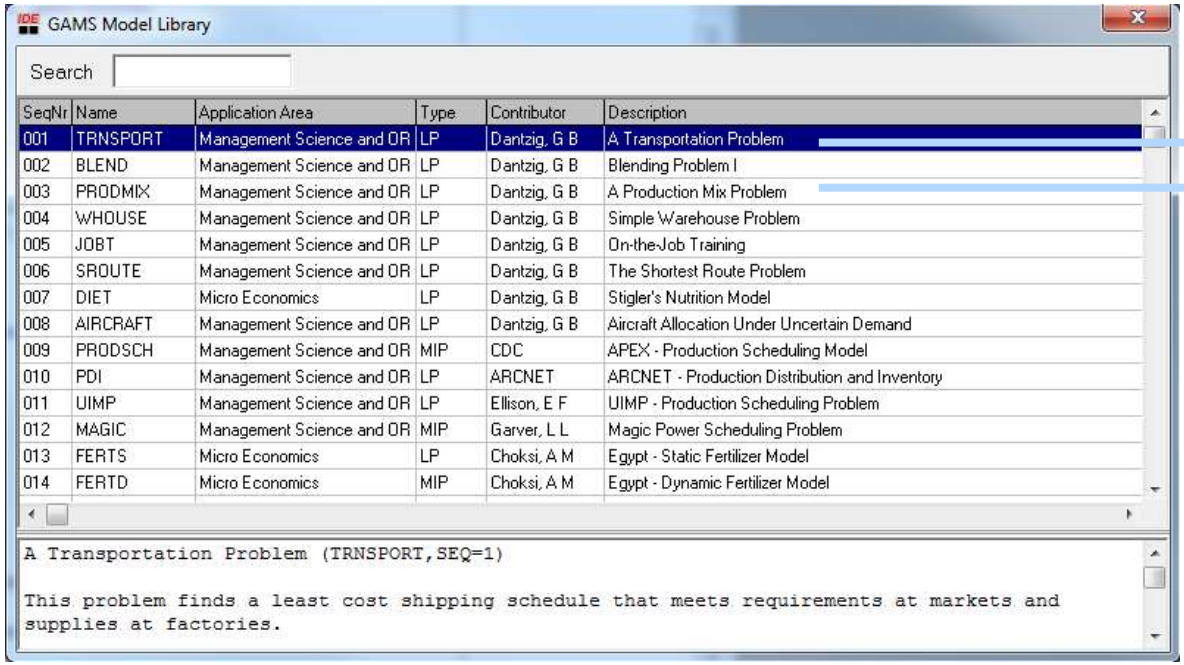
Representation in GAMS (Input)

```
Sets  desk / d1, d2, d3, d4 /
      shop / carpentry, finishing /

Table labor(shop,desk)  labor requirements (man-hours)

               d1    d2    d3    d4
  carpentry     4     9     7    10
  finishing     1     1     3    40

Parameters

   caplim(shop)  capacity (man hours) / carpentry = 6000, finishing = 4000 /
   price(desk)   per unit sold ($)    / d1 = 12, d2 = 20, d3 = 18, d4 = 40 /

Variables

   mix(desk)  mix of desks produced (number of desks)
   profit     total profit                    ( $ )

Positive Variable  mix

Equations

   cap(shop) capacity constraint            (man-hours)
   ap        accounting: total profit             ( $ );

cap(shop).. sum(desk, labor(shop,desk)*mix(desk)) =l= caplim(shop);

ap..  profit =e= sum(desk, price(desk)*mix(desk));

Model pmp product mix problem / all /; Solve pmp maximizing profit using lp;
```

Imperial College London

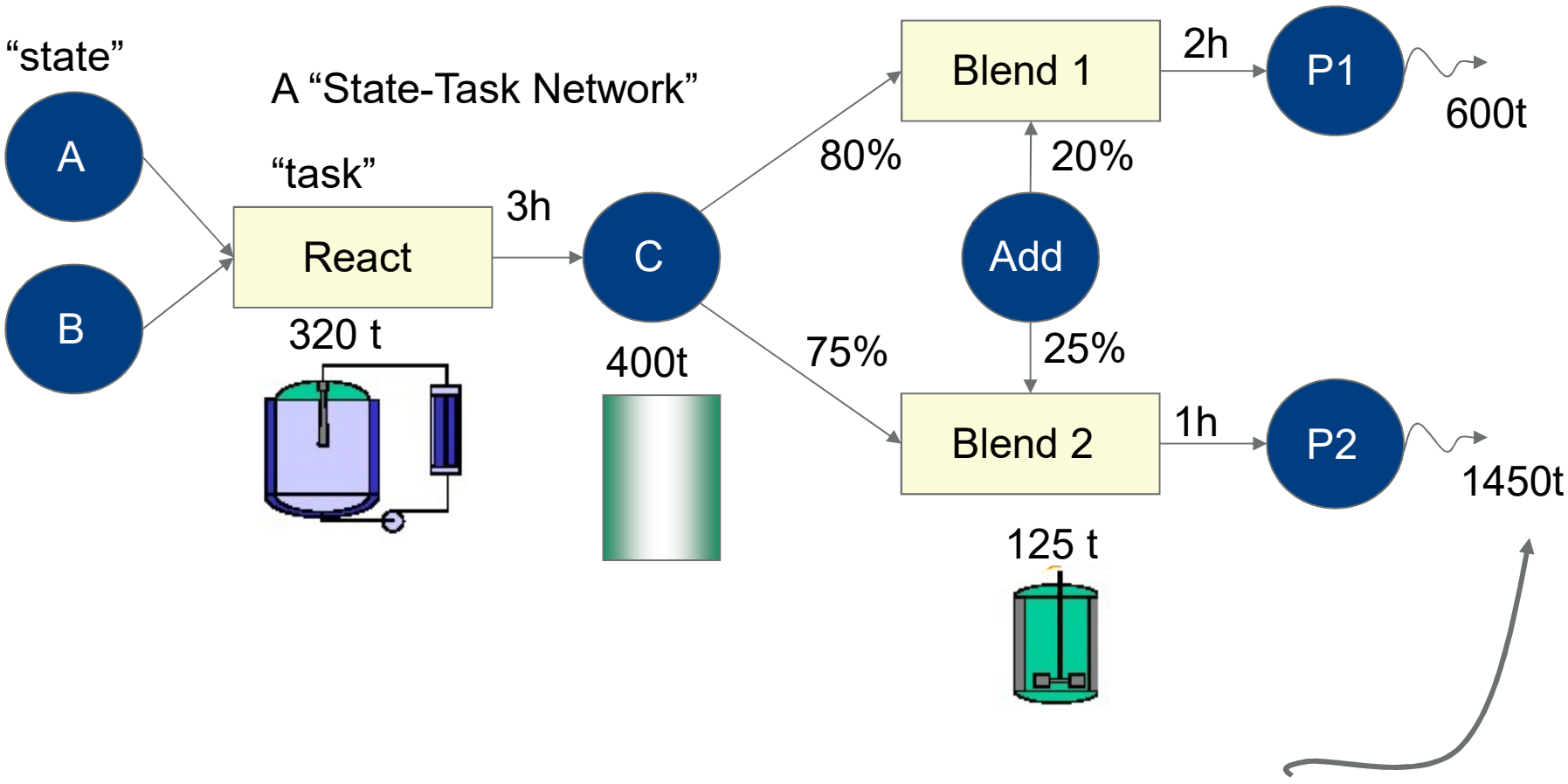
Getting Help

- GAMS User Guide
- Expanded GAMS Guide (McCarl)
- Example from Model Library



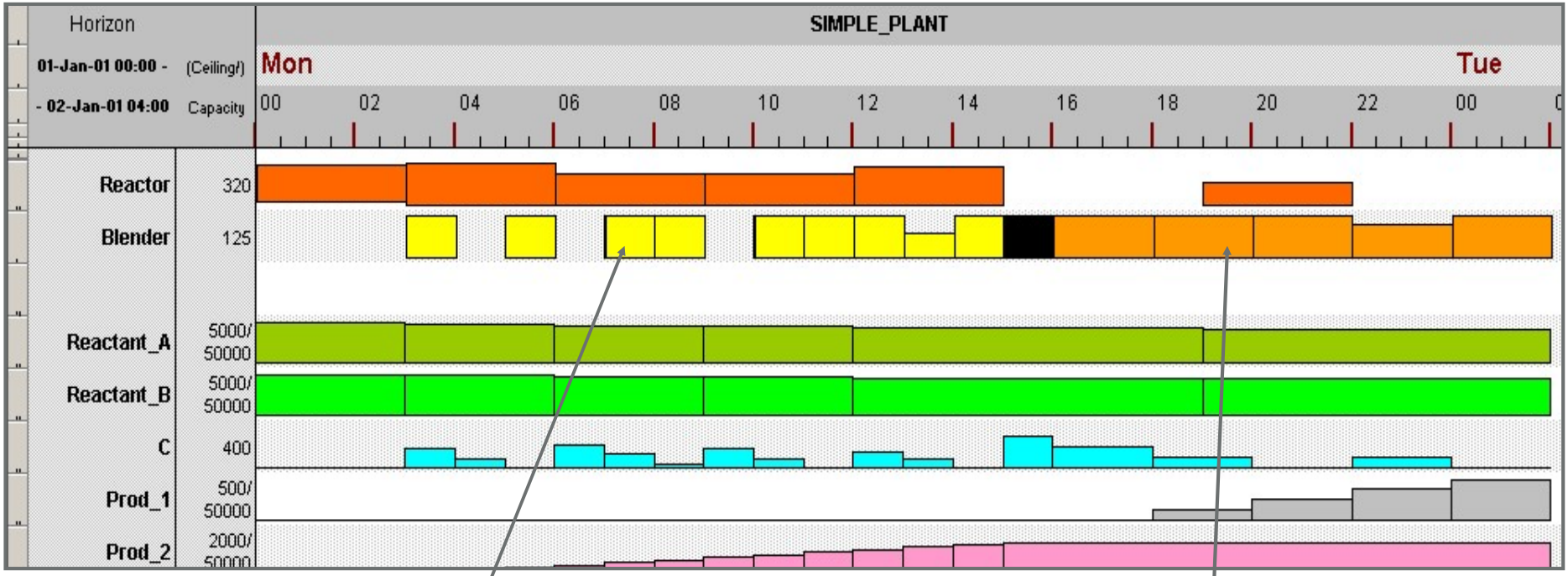
Imperial College London

Consider this problem



Imperial College
London

Answer: 26 hours



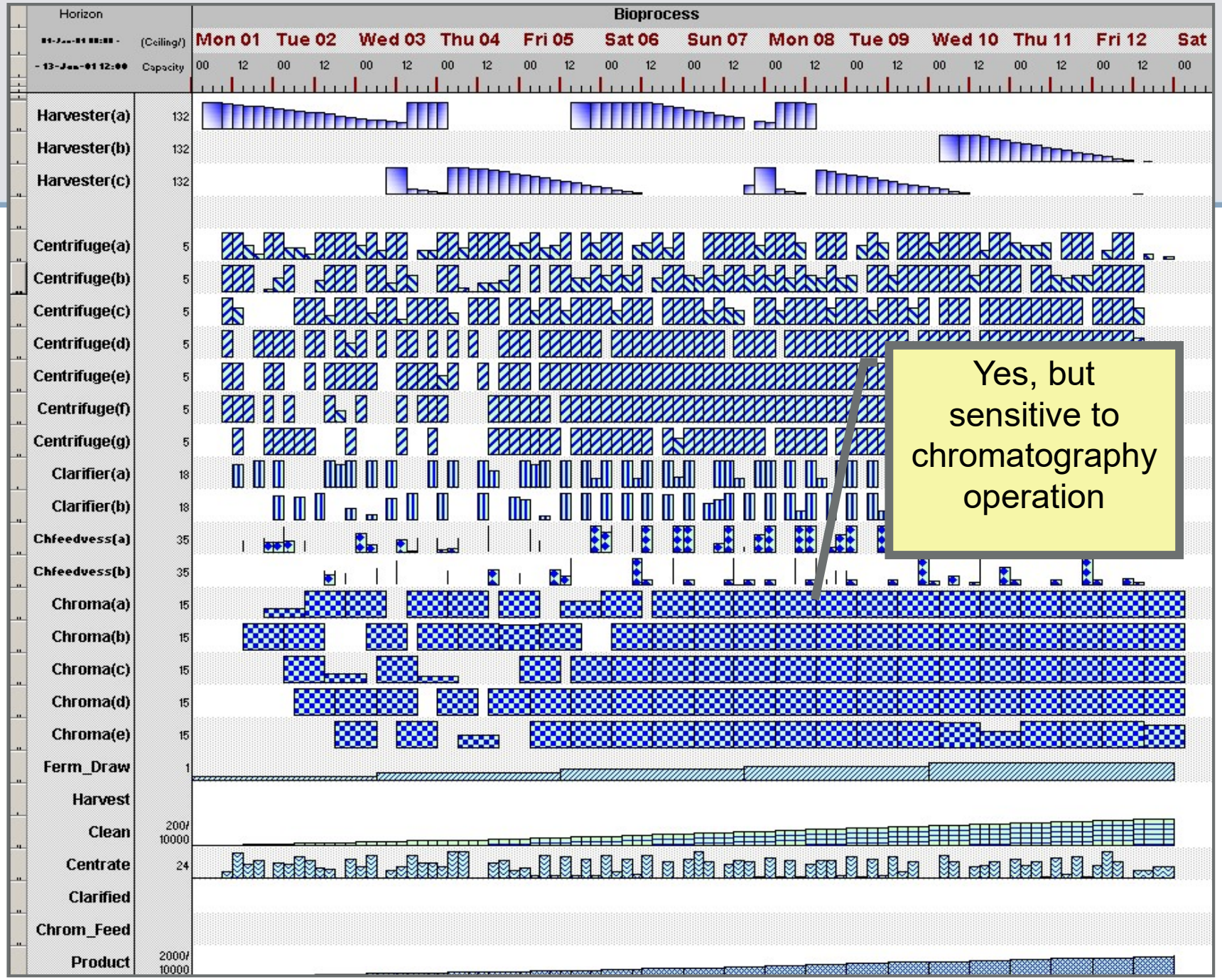
Imperial College
London

Will a bioprocess design work?

Interconnected process steps

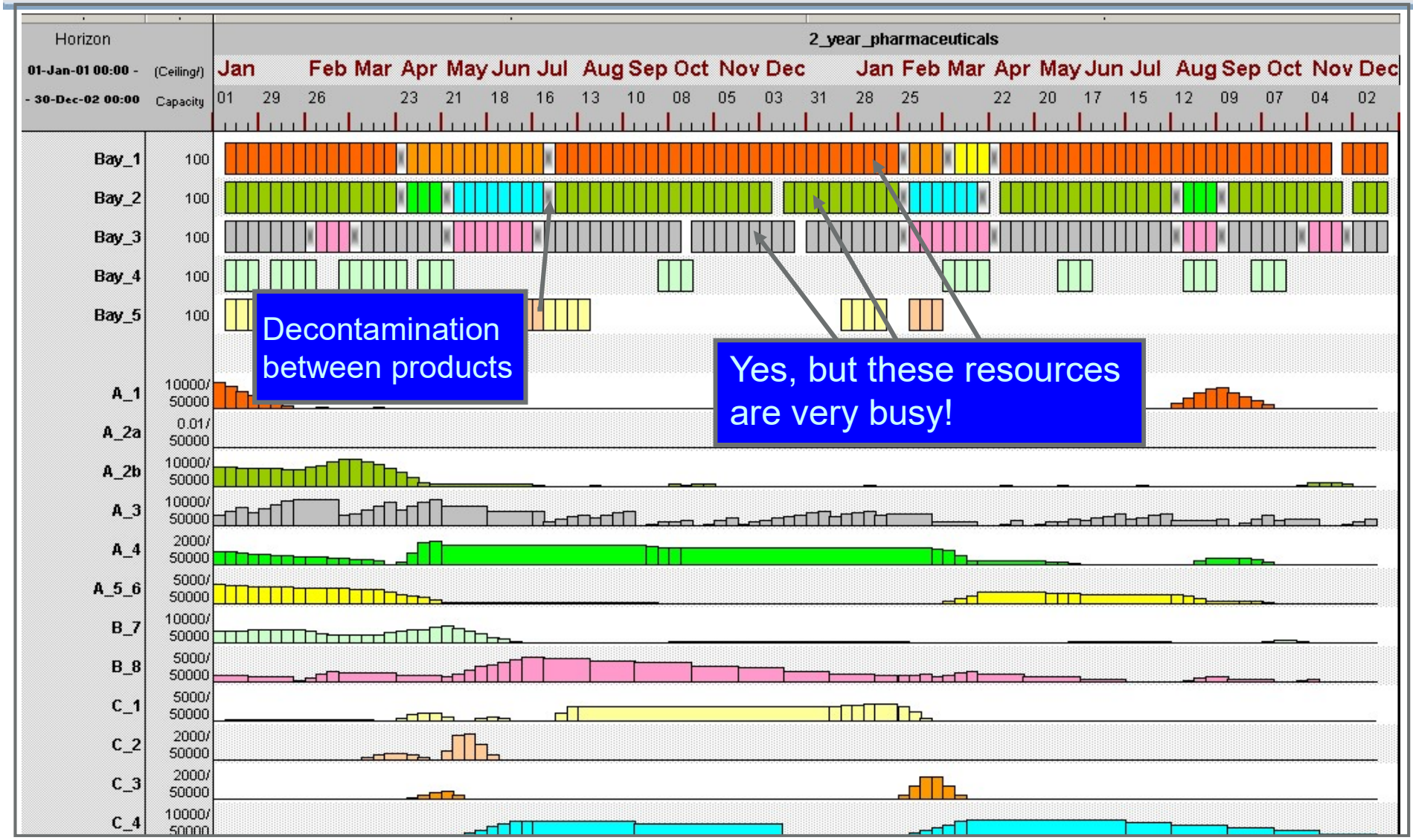
Complex purification

Many equipment items


Horizon
(Ceiling/)
Capacity
Bioprocess
Mon 01
Tue 02
Wed 03
Thu 04
Fri 05
Sat 06
Sun 07
Mon 08
Tue 09
Wed 10
Thu 11
Fri 12
Sat
Harvester(a) 132
Harvester(b) 132
Harvester(c) 132
Centrifuge(a) 5
Centrifuge(b) 5
Centrifuge(c) 5
Centrifuge(d) 5
Centrifuge(e) 5
Centrifuge(f) 5
Centrifuge(g) 5
Clarifier(a) 18
Clarifier(b) 18
Chfeedvess(a) 35
Chfeedvess(b) 35
Chroma(a) 15
Chroma(b) 15
Chroma(c) 15
Chroma(d) 15
Chroma(e) 15
Ferm_Draw 1
Harvest
Clean 200/ 10000
Centrate 24
Clarified
Chrom_Feed
Product 2000/ 10000
Yes, but sensitive to chromatography operation

Imperial College
London

Can this site make enough of these new pharmaceuticals?



Imperial College London

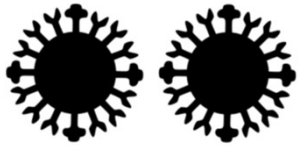
Conclusions

- We are working together with developing country manufactures to improve their vaccine production processes;
- We are designing and implementing new vaccine production technologies;
- We are doing multi-level modelling using the following tools:
    - gPROMS (unit operation level);
    - SuperPro Designer (process flow sheet level);
    - GAMS (supply chain level).

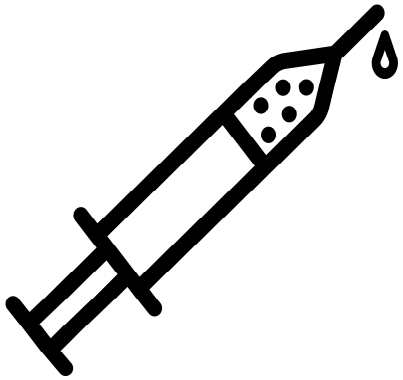
Imperial College London

Zoltan Kis, Maria Papathanasiou, Cleo Kontoravdi and Nilay Shah

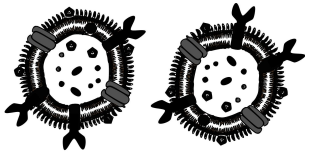
RNA



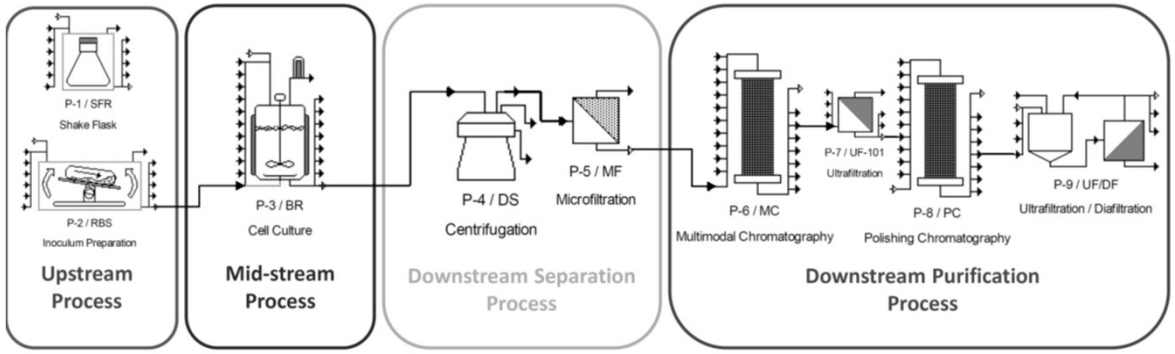
ADDomer



h Yeast



GMMA



https://www.imperial.ac.uk/future-vaccine-hub