

# **BIOMENG 261**

## **TISSUE AND BIOMOLECULAR ENGINEERING**

*Module I: Reaction kinetics and systems biology*

*Oliver Maclaren*  
*oliver.maclaren@auckland.ac.nz*

# MODULE OVERVIEW

Reaction kinetics and systems biology (*Oliver Maclaren*)  
**[11 lectures/3 tutorials/2 labs]**

*1. Basic principles: modelling with reaction kinetics* [4 lectures]

Conservation, directional and constitutive principles. Mass action. Enzyme kinetics. Enzyme regulation. Mathematical/graphical tools for analysis and fitting.

*2. Systems biology I: signalling and metabolic systems* [2 lectures]

Overview of systems biology. Modelling signalling systems using reaction kinetics. Introduction to parameter estimation. Modelling metabolic systems using reaction kinetics. Flux balance analysis and constraint-based methods.

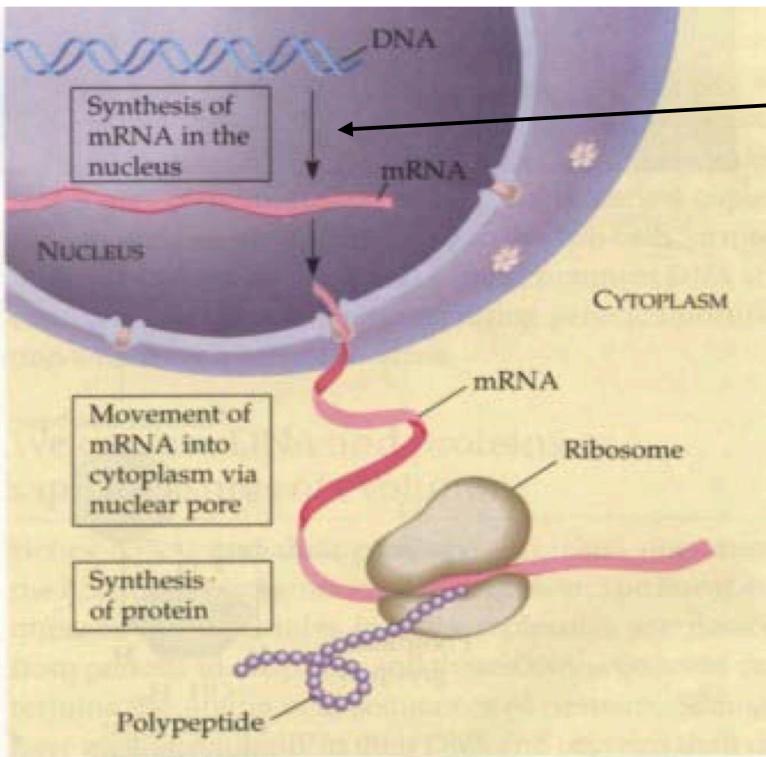
*3. Systems biology II: genetic systems* [3 lectures]

Modelling genes and gene regulation using reaction kinetics. Gene regulatory networks, transcriptomics and analysis of microarray data.

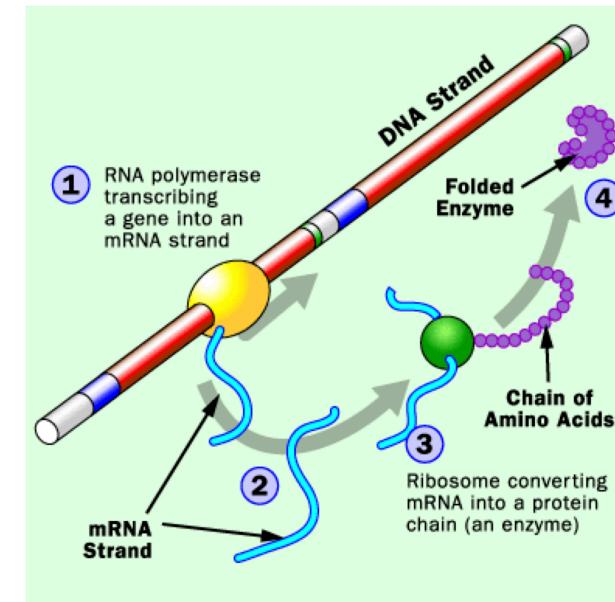
# LECTURE 9: MODELLING GENE EXPRESSION AND REGULATION

- Using 'reaction' language to describe gene expression and regulation
- In particular: transcription and its regulation
- Gene regulatory states and occupancy probabilities/fractions
- Using quasi-equilibrium gene-state model to derive overall constitutive model for transcriptional flux

# GENE EXPRESSION AND REGULATION

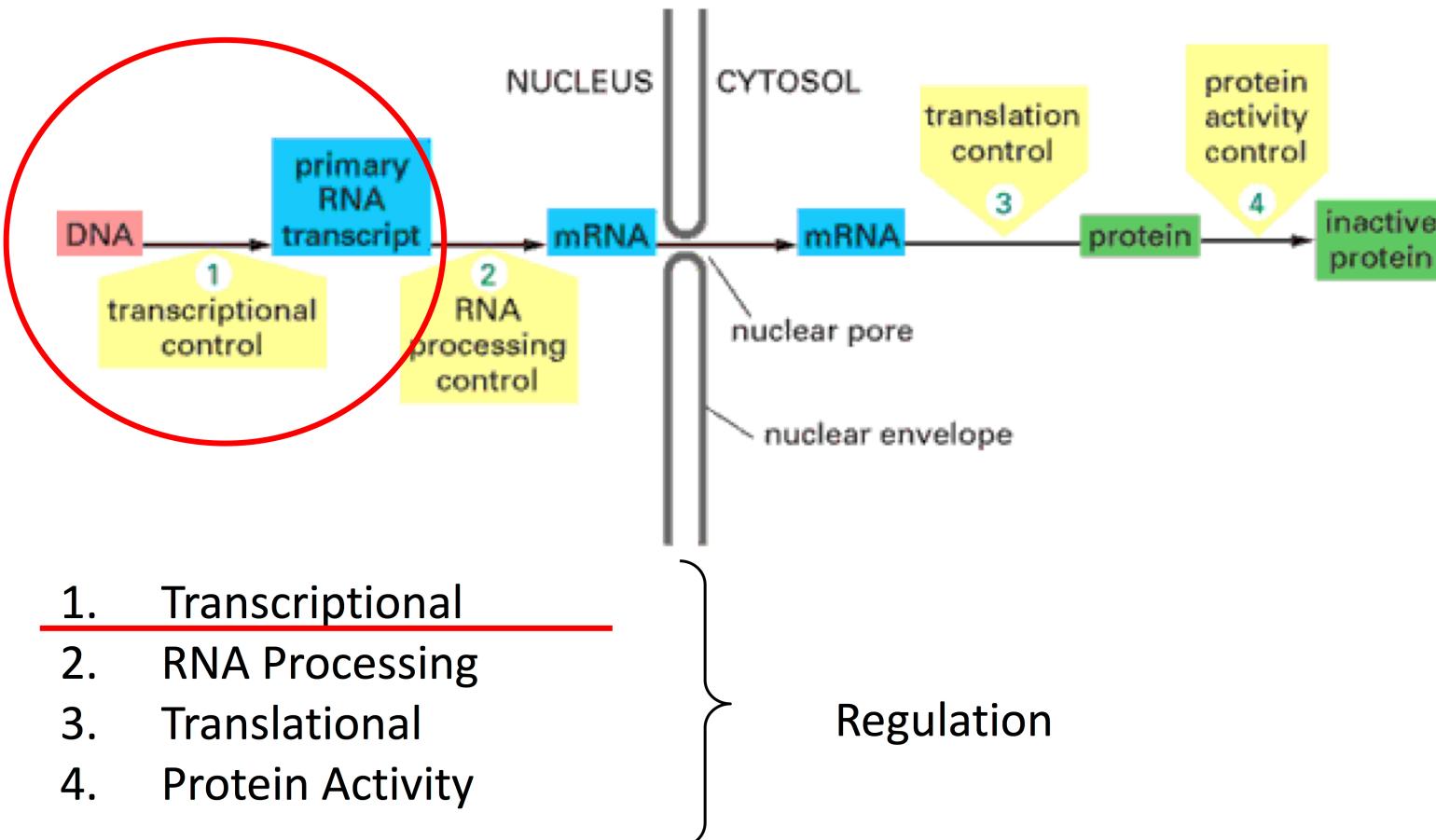


RNA Polymerase

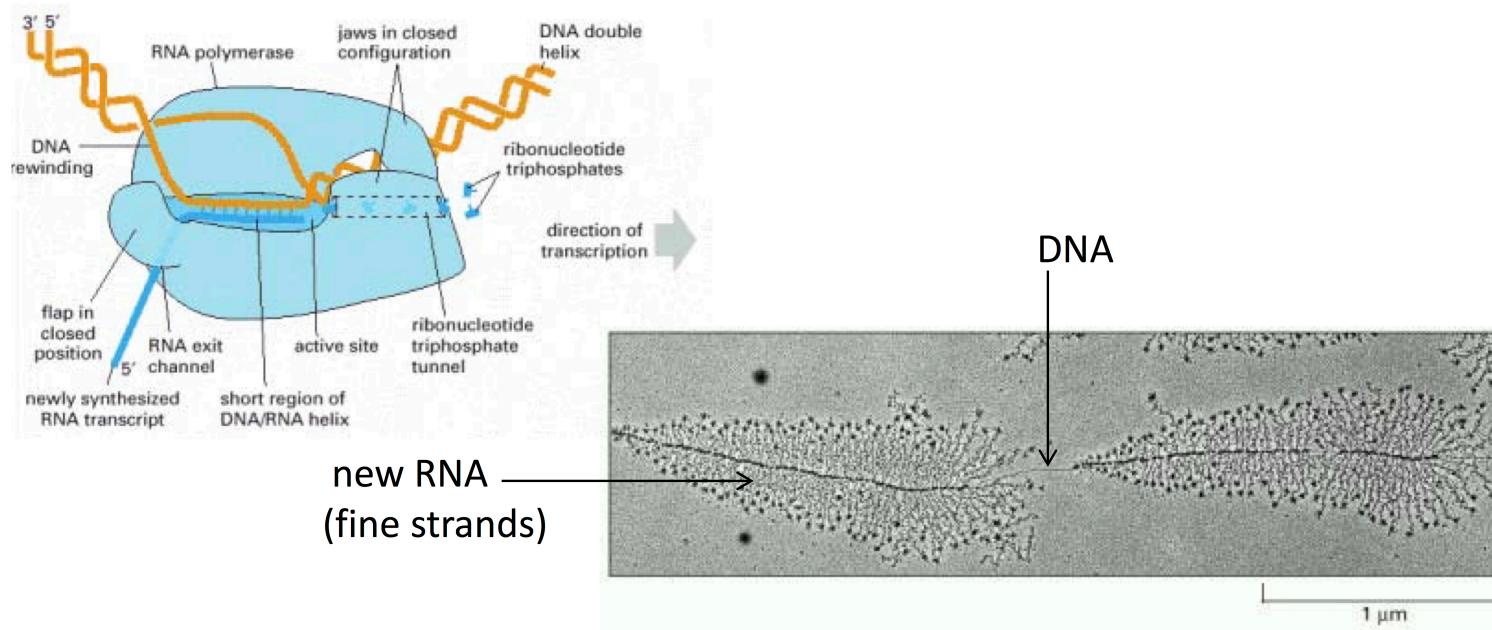


DNA  $\xrightarrow[\text{(transcription)}]{\text{RNA polymerase}}$  mRNA  $\xrightarrow[\text{(translation)}]{\text{Ribosomes}}$  Protein

# GENE EXPRESSION AND REGULATION



# FOCUS: GENE TRANSCRIPTION



Many molecules of RNA polymerase (beads on DNA) simultaneously transcribing each of two adjacent genes

Images from *Molecular Biology of the Cell* (4ed), "From DNA to RNA", Fig 6-8 and 6-9.  
Online at [www.ncbi.nlm.nih.gov/books/NBK26887/](http://www.ncbi.nlm.nih.gov/books/NBK26887/)

# TRANSCRIPTION REGULATION TYPES

*Gene transcription* is regulated by various *transcription factors*.

These can be

- *Activators* (increase transcription)
- *Repressors* (decrease transcription)

When a gene codes for its *own* activator/repressor we call this positive/negative *autoregulation*.

## OVERALL BALANCE EQUATIONS

As usual, we can write overall *conservation equations*

$$\frac{dR}{dt} = v_{\text{transcription}} - v_{\text{Rdeg}}$$

$$\frac{dP}{dt} = v_{\text{translation}} - v_{\text{Pdeg}}$$

where  $R$  is mRNA,  $P$  is protein/product.

Note: using  $v$  instead of  $J$  just for consistency with typical approaches. Same basic thing - a *flux*.

## FOCUS: TRANSCRIPTION FLUX

Our goal here is to derive a *constitutive equation* for  $v_{\text{transcription}}$  in terms of underlying gene '*regulatory states*'.

- Essentially the same idea as enzyme kinetics: more *detailed model + equilibrium assumption* to get overall flux expression.

# GENE REGULATORY STATES

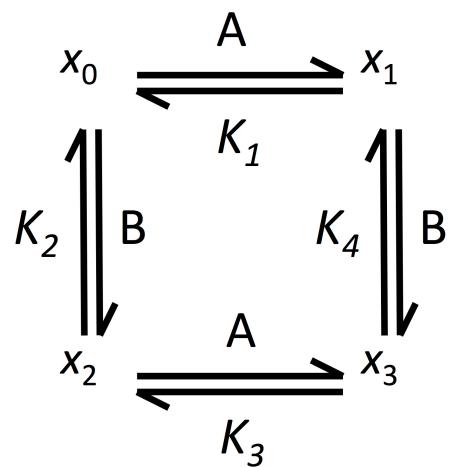
We view the gene as existing in a *number of states with rapid transitions between them.*

Each state (potentially) contributes to the *overall flux*.

Rather than 'concentrations' we use state *fractions* i.e. state *occupancy probabilities* and then use these to average the flux contributions.

We use the *equilibrium approximation* to determine the underlying probabilities to use in averaging.

# GENE REGULATORY STATES AND FLUXES



where

$$x_0 + x_1 + x_2 + x_3 = 1$$

are the *occupancy probabilities/state fractions*...which we solve for and plug into...

# OVERALL FLUX CONSTITUTIVE EQUATION

...the *overall constitutive equation* for transcriptional flux

$$v_{\text{transcription}} = x_0 v_0 + x_1 v_1 + x_2 v_2 + x_3 v_3$$

i.e.

$$v_{\text{transcription}} = \sum_{s=0}^{N_s-1} x_s v_s$$

where the choice of the  $v_i$  depends on whether the TFs are activators/repressors etc.

This plugs back into...

## OVERALL BALANCE EQUATIONS

$$\frac{dR}{dt} = v_{\text{transcription}} - v_{\text{Rdeg}}$$

$$\frac{dP}{dt} = v_{\text{translation}} - v_{\text{Pdeg}}$$

where  $R$  is mRNA,  $P$  is protein/product.

# **EXAMPLES**

See handout.