Membrane Interactions

Luca Cardelli

Microsoft Research

CSSB School, Rovereto 2004-04-23

http://www.luca.demon.co.uk

Biological *Systems*

The emerging area of *Systems Biology*: interdisciplinary study of relationships and interactions of biological components

"The types of biological information (DNA, RNA, protein, protein interactions, biomodules, cells, tissues, etc.) also have their individual elements (e.g. specific genes or proteins) and the relationships of these with respect to one another and the elements of other types of biological information must be determined and, once again, all of this information integrated to obtain a view (model) of the system as a whole."

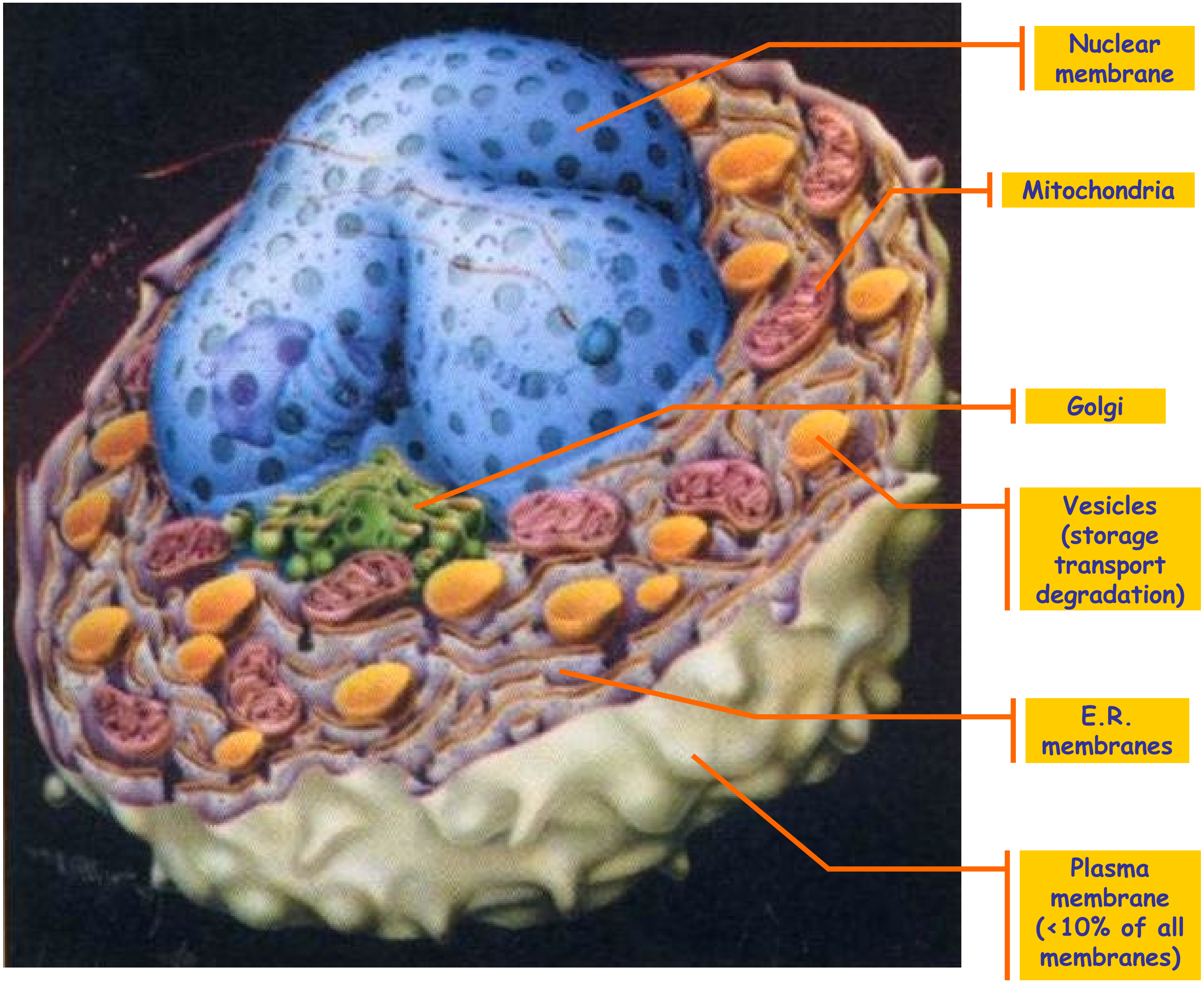
"Hence the importance of high-throughput facilities with global capacities (e.g., measure all genes or all proteins) and a strong computational infrastructure …"

"Discovery science must be integrated with hypothesis-driven science for the integrated global analysis of systems."

http://www.systemsbiology.org/

Eukaryotic Cell

Membranes everywhere



Nuclear membrane

Mitochondria

Golgi

Vesicles (storage transport degradation)

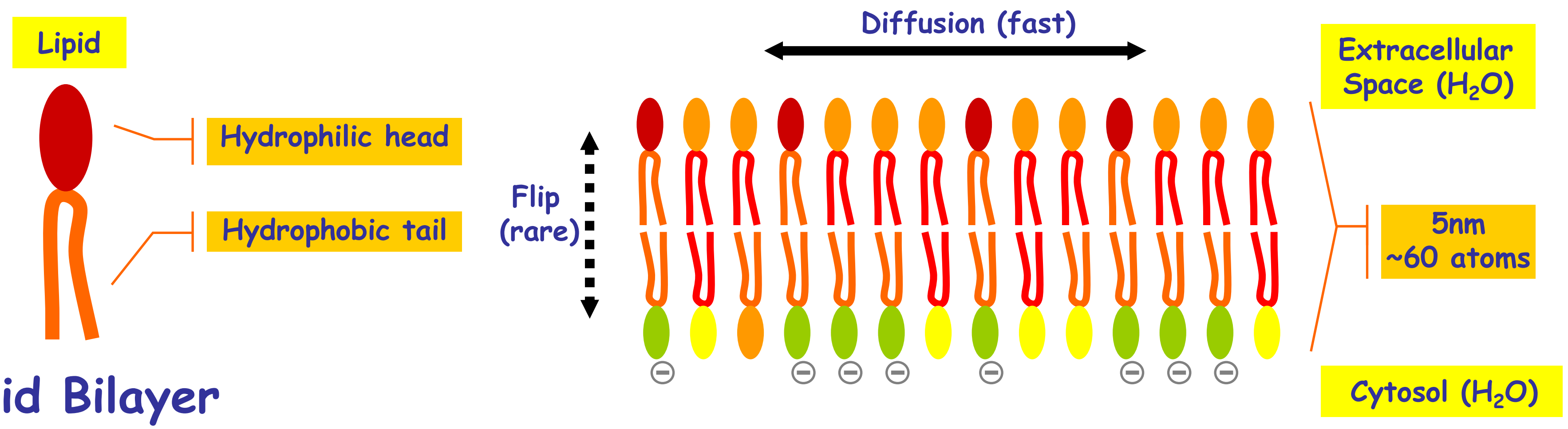
E.R. membranes

Plasma membrane (<10% of all membranes)

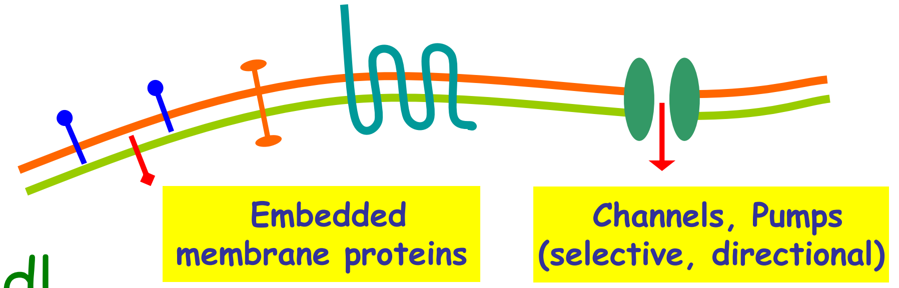
Importance of Membranes

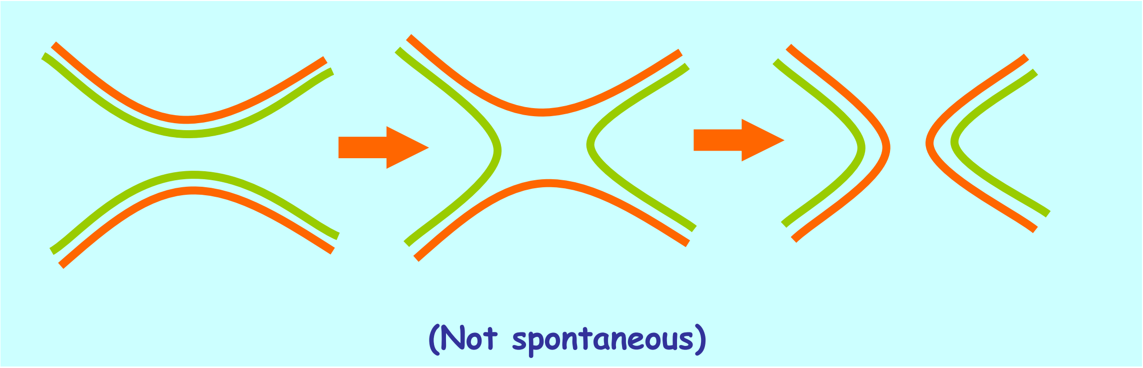
- Many cellular processes involve membranes. It's *very far* from a "chemical soup":
  - For a cell to function properly, each of its numerous proteins must be localized to the correct cellular membrane or aqueous compartment. [MCB p.675]
- What is the structure and dynamics of these complex configurations of membranes?

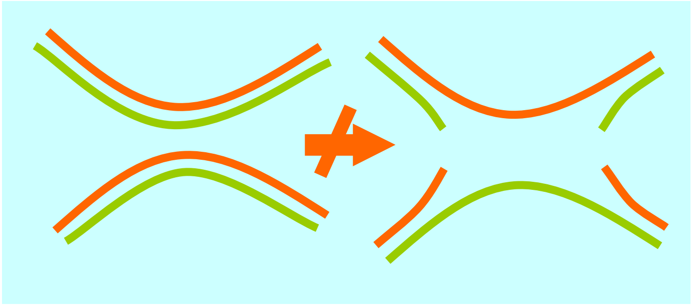
Membranes are Oriented 2D Surfaces



Lipid Bilayer
Self-assembling
Largely impermeable
Asymmetrical (in real cells)
With embedded proteins
A 2D fluid inside a 3D fluid!


Embedded membrane proteins
Channels, Pumps (selective, directional)

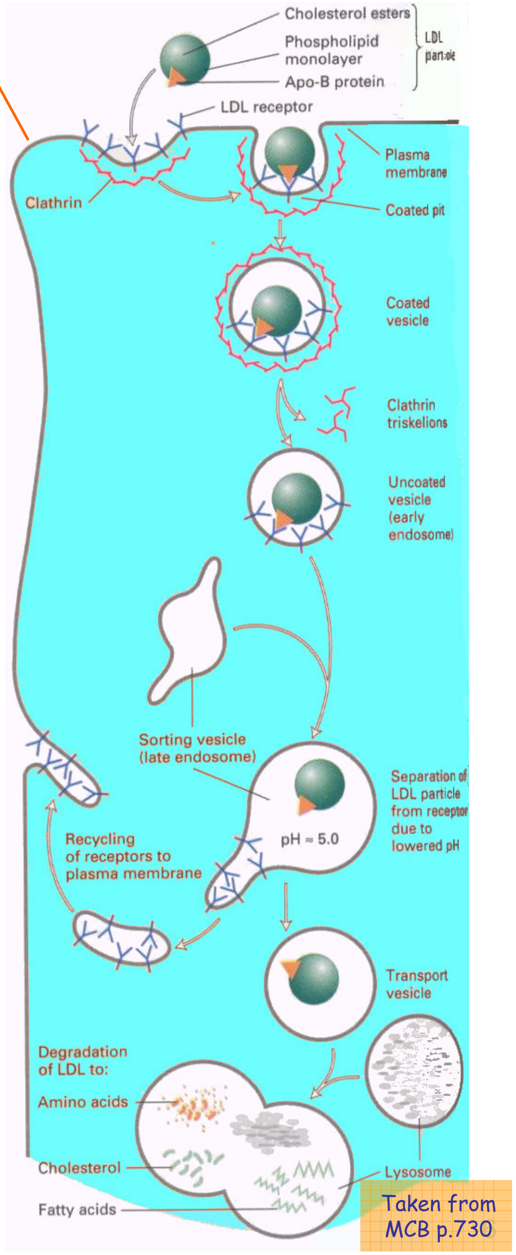




A Biological Algorithm

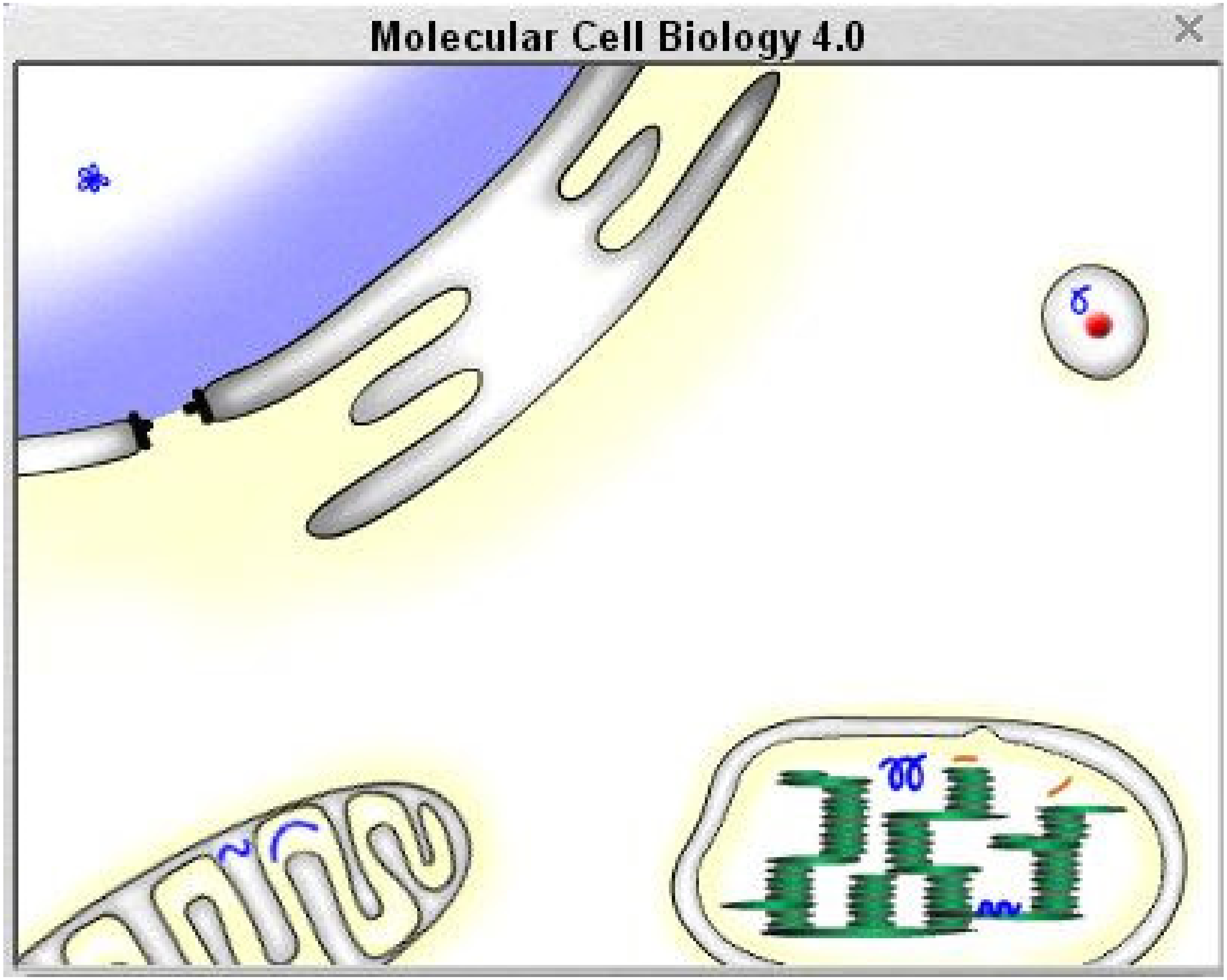
- LDL-Cholesterol Degradation
  - A cast of many thousands (molecules) just to get one molecule from A to B.
  - Membranes are key to the algorithm, we want to model *them*, not their individual millions of molecules.
- Some very fancy chemistry
  - But its "purpose" is to reliably implement a specific sequence of discrete steps.

Lipid bilayer



Taken from MCB p.730

Dynamic Compartments



Taken from MCB CD

Aims

- Describing biological *processes* in order to study and understand their dynamic evolution
  - More precisely than pictures and informal diagrams.
  - Writing bioalgorithms in something close to a language.
  - For precision, analysis, simulation, storage, search…
- Abstraction/modeling options
  - Start too low ⇒ get lost in a mess of details.
  - Start too high ⇒ miss too much behavior.
  - Often to model different abstraction levels:
    - When nature "cheats".
    - When different regimes/timescales interact.
- Evolving approach - Common technology
  - Molecular Reactions, using process calculi (BioSPi)
  - Reactions + Membranes (BioAmbients)
  - Reactions *on* Membranes (*Brane Calculi*)
- Focus on (notations for) membranes
  - But they need to be understood in the context of (notations for) the other cellular subsystems…

Purpose: survey not molecular biology, but its *notations*.

The Functional Architecture(*) of Biological Cells

Molecular biology notations embody:

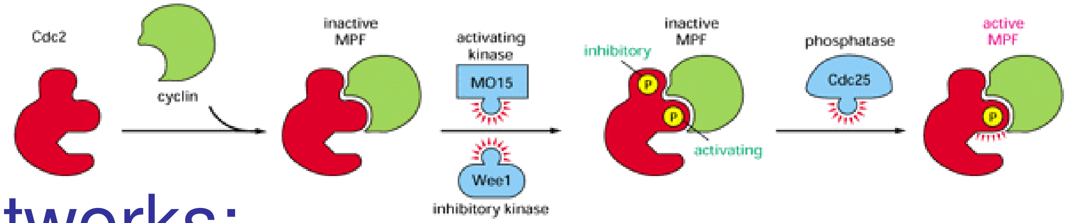
Three Virtual Machines

Biochemical Networks - The Protein Machine
Gene Regulatory Networks - The Gene Machine
Transport Networks - The Membrane Machine

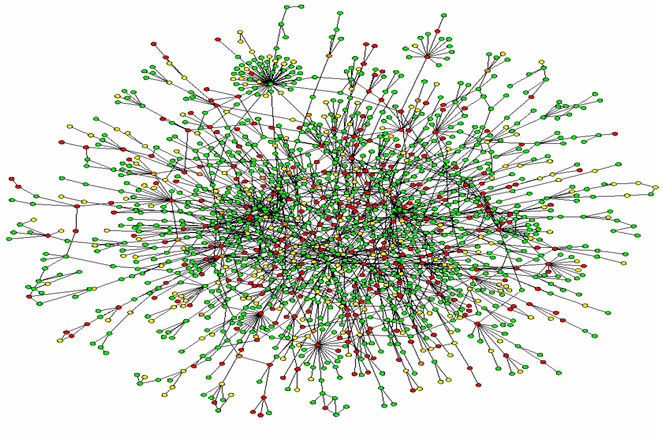
(*) Major functional subsystems and how they fit together

1. The Protein Machine

- Complex folded-up shapes that:
  - Fit together, stick, unstick.
  - Excite/unexcite, warp each other.
  - Bring together, catalyze, transform materials.
  - Form complex aggregates and networks.

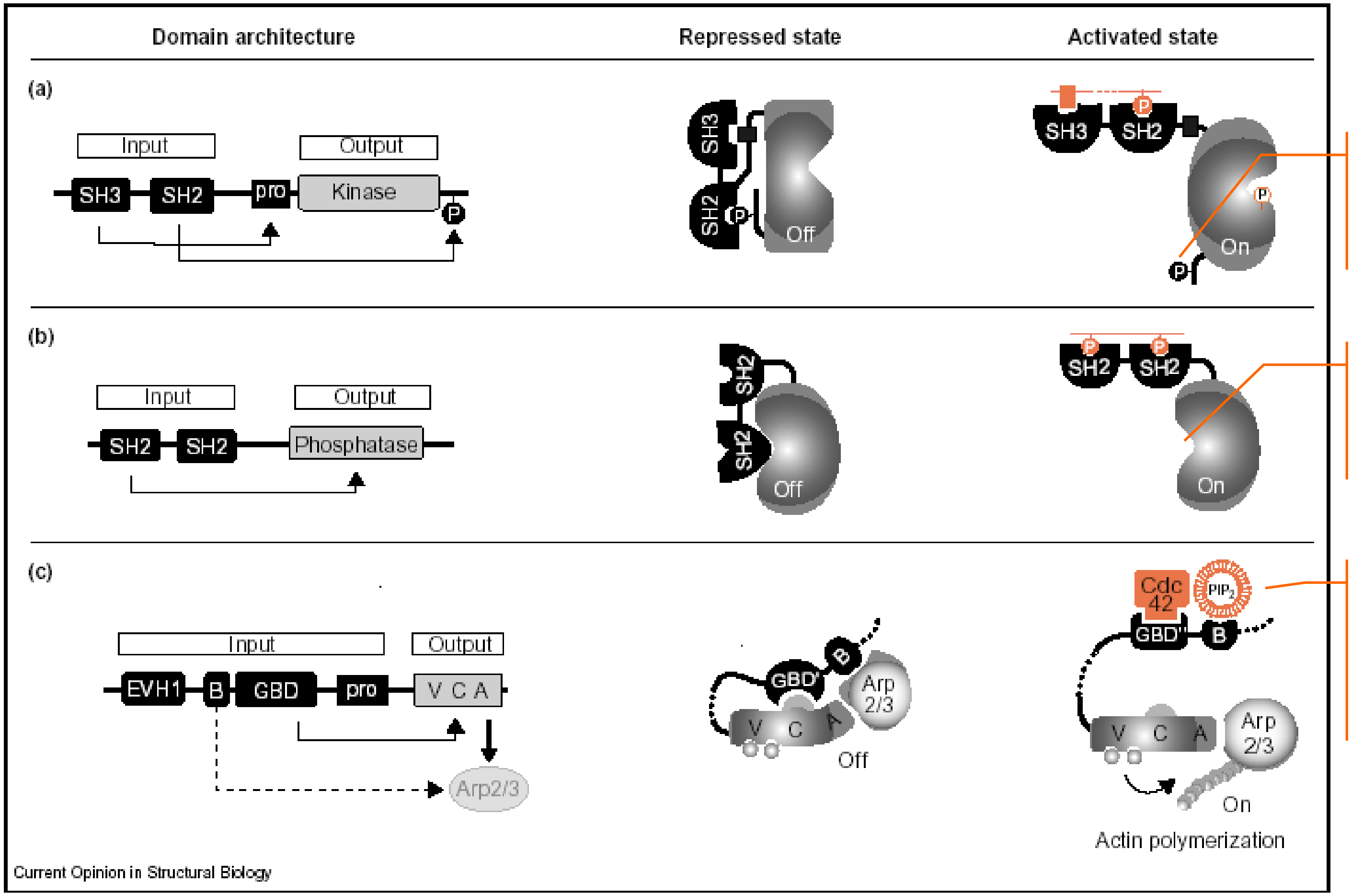


- Mapping out such networks:
  - In principle, it’s “just” a very large set of chemical equations.
  - Notations have been developed to summarize and abstract.



An actual molecular interaction network.
(Nodes are distinct protein kinds,
arcs mean that two kinds of proteins interact.)

Some Allosteric Switches



Current Opinion in Structural Biology

Domain architecture and autoinhibitory interactions in modular switch proteins. (a) Src family kinases contain N-terminal SH3 and SH2 domains, and a kinase domain flanked by intramolecular SH3-binding and SH2-binding sites (when the C-terminal motif tyrosine is phosphorylated by Csk). The crystal structures of several family members show that both intramolecular domain interactions function in concert to lock the kinase in an inactive conformation. Activating stimuli (red) include external SH2 or SH3 ligands. After initial activation, the kinase is maintained in an active state by autophosphorylation of its activation loop. (b) SHP-2 phosphatase contains two SH2 domains and a phosphatase domain. The crystal structure of the phosphatase shows that the N-terminal SH2 domain participates in an autoinhibitory interaction that directly blocks the phosphatase active site. Binding of external SH2 ligands activates by disrupting the autoinhibitory interaction. (c) N-WASP contains an Enabled VASP homology 1 (EVH1) domain, a B motif, a GBD, a proline-rich segment (pro) and an output region (VCA) that alone binds the Arp2/3 complex and stimulates its actin nucleation activity. The B and GBD motifs are required to repress activity and, by current models, are thought to participate in intracomplex interactions (only the structure of the GBD intramolecular complex for WASP is known). GTP-bound Cdc42 and PIP_2 synergistically activate N-WASP.

Allosteric ("other shape") reactions modify accessibility.

Kinase

= donates phosphate P
= phosphorilates other proteins

Phosphatase

= accepts phosphate P
= dephosphorilates other proteins

Logical AND

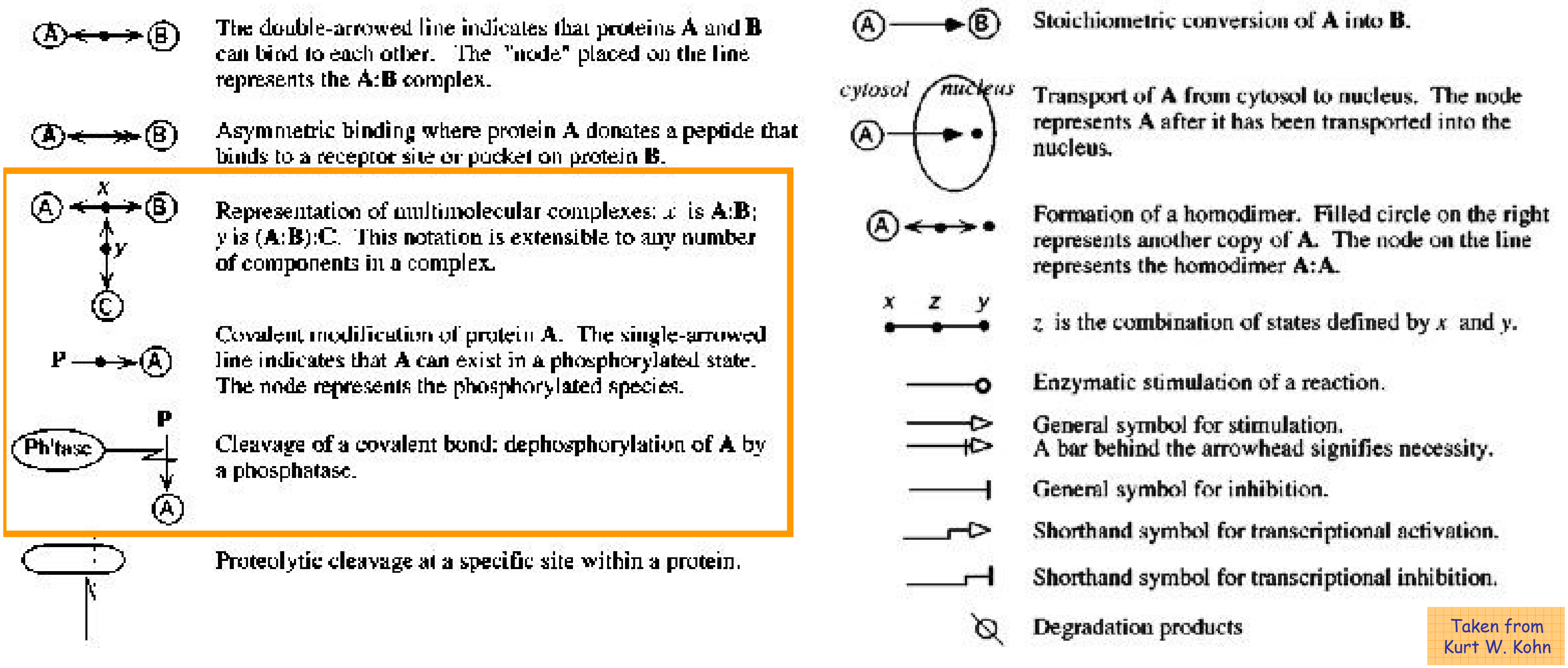
at equal concentrations of the individual input stimuli, activation is much higher if both stimuli are present

"Phosphatase Kinase Kinase" = a kinase that activates a kinase that activates a phosphatase that deactivates a protein.

Humans have the same number of modular protein *domains* (building blocks) as worms, but twice the number of multi-domain proteins.

Taken from Wendell Lim

MIM: Molecular Interaction Maps (Kohn)



The double-arrowed line indicates that proteins **A** and **B** can bind to each other. The "node" placed on the line represents the **A:B** complex.

Asymmetric binding where protein **A** donates a peptide that binds to a receptor site or pocket on protein **B**.

Representation of multimolecular complexes: *x* is **A:B**; *y* is (**A:B**):**C**. This notation is extensible to any number of components in a complex.

Covalent modification of protein **A**. The single-arrowed line indicates that **A** can exist in a phosphorylated state. The node represents the phosphorylated species.

Cleavage of a covalent bond: dephosphorylation of **A** by a phosphatase.

Proteolytic cleavage at a specific site within a protein.

Stoichiometric conversion of **A** into **B**.

Transport of **A** from cytosol to nucleus. The node represents **A** after it has been transported into the nucleus.

Formation of a homodimer. Filled circle on the right represents another copy of **A**. The node on the line represents the homodimer **A:A**.

z is the combination of states defined by *x* and *y*.

Enzymatic stimulation of a reaction.

General symbol for stimulation.
A bar behind the arrowhead signifies necessity.

General symbol for inhibition.

Shorthand symbol for transcriptional activation.

Shorthand symbol for transcriptional inhibition.

Degradation products

Taken from
Kurt W. Kohn

The p53-Mdm2 and DNA Repair Regulatory Network

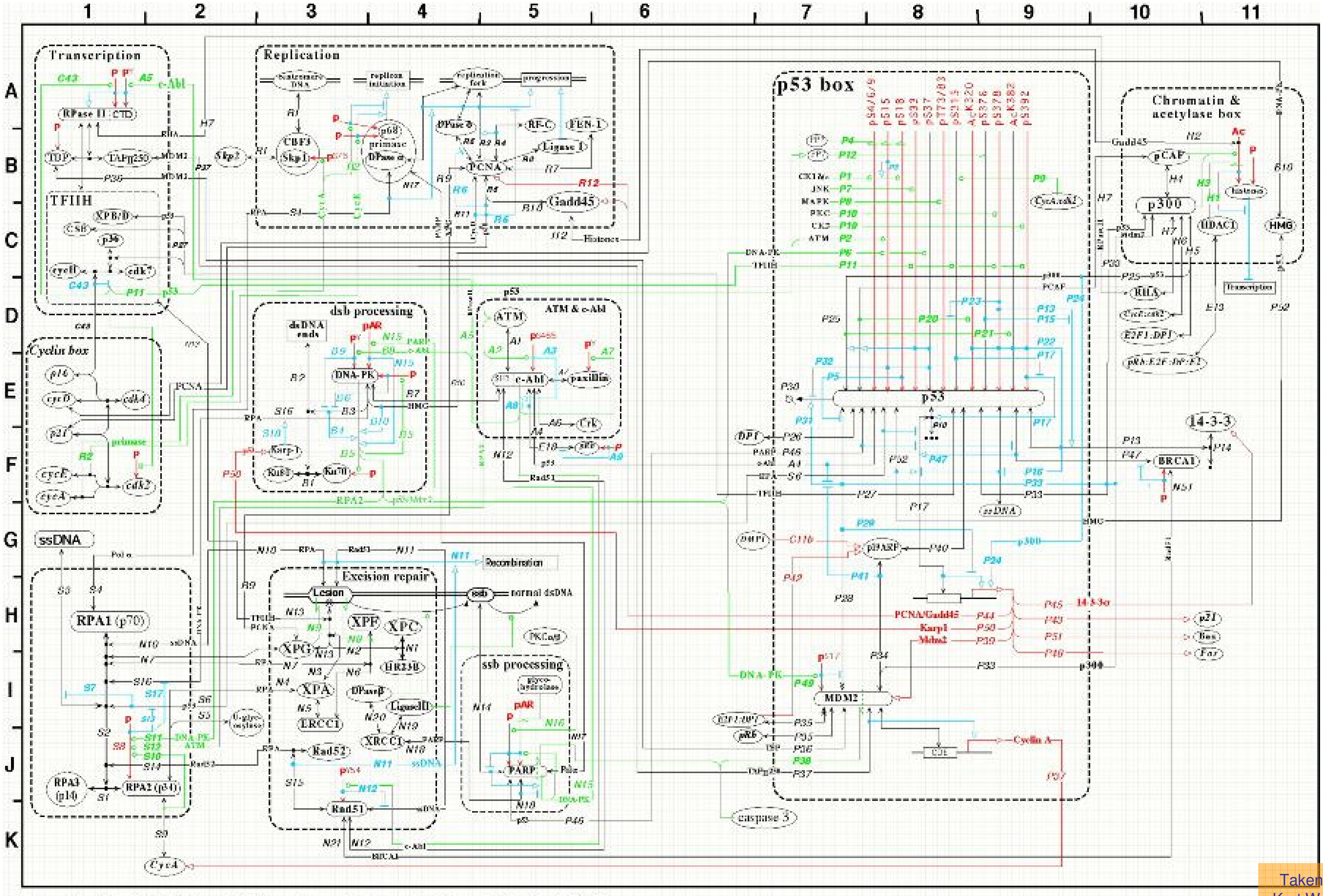
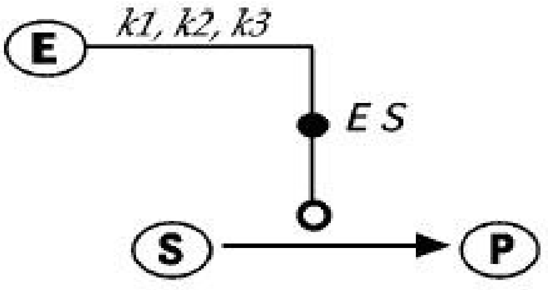


Figure 6B: The p53-Mdm2 and DNA repair regulatory network (version 2p - May 19, 1999)

Taken from Kurt W. Kohn

Kohn Diagrams



$$E + S \underset{k2}{\overset{k1}{\rightleftarrows}} E\,S \xrightarrow{k3} P$$

FIG. 3. Simple one-way enzymatic reaction. (If there is an energy source, such as ATP hydrolysis, it can be omitted when ATP concentration is not an important factor.) In explicit formulations, the reaction identifiers or rate constant designations can be placed on the enzyme reaction line, and the node ES can identify the enzyme-substrate species.

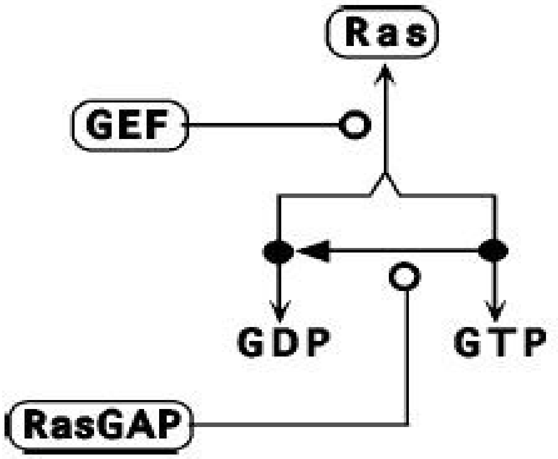


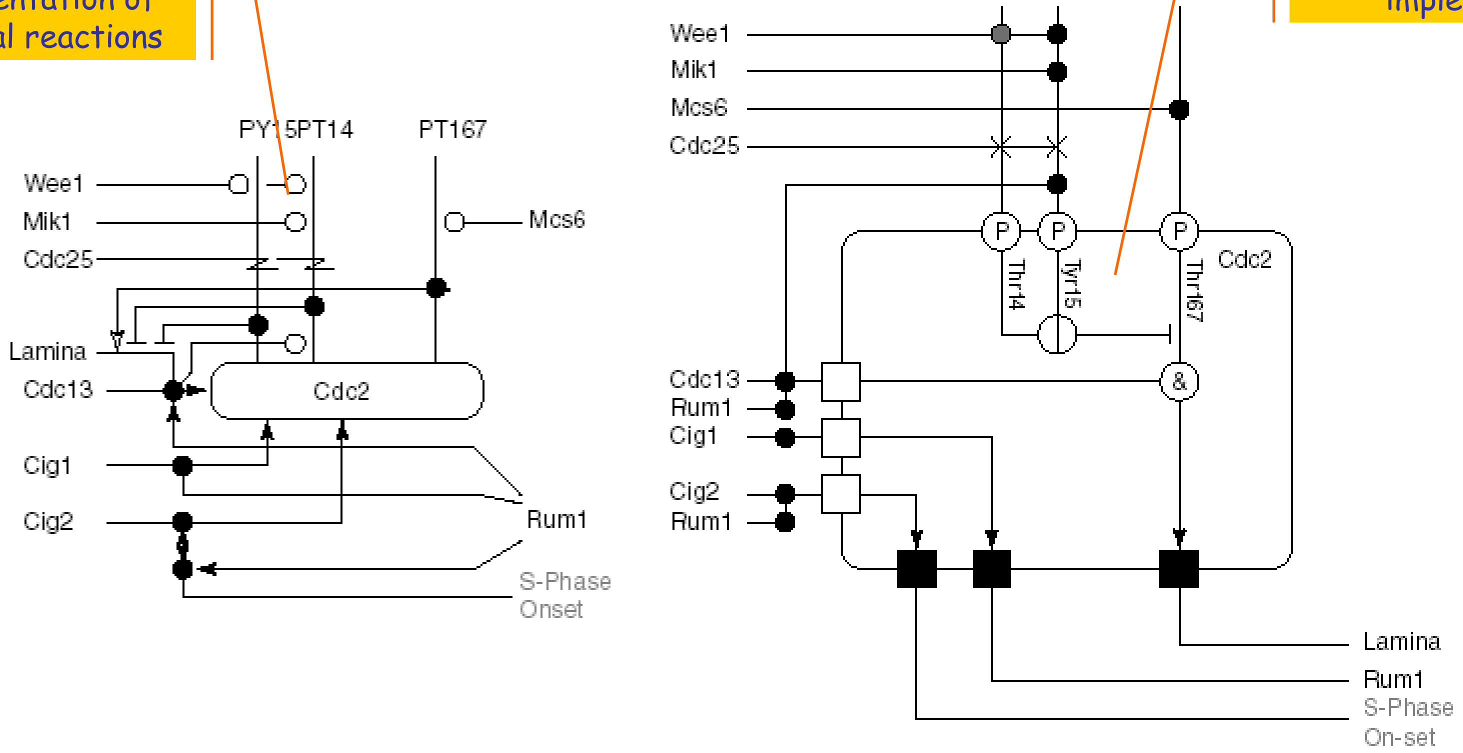
FIG. 4. Interconversions between the GTP- and GDP-bound states of Ras. (1) GDP and GTP compete with each other for binding to a site on Ras (this binding is only slowly reversible). (2) GEF (guanine nucleotide exchange factor) facilitates the binding or dissociation of GDP or GTP (the concentration of GTP normally far exceeds that of GDP). (Implicit is the reversible binding between GEF and Ras which opens the binding site for GDP/GTP exchange.) (3) Ras has an intrinsic GTPase activity that slowly converts bound GTP to bound GDP (stoichiometric conversion arrow points from the node representing Ras.GTP to the node representing Ras.GDP). (4) RasGAP (a GTPase activating protein) enhances the GTPase activity of Ras. (Implicit is the reversible enzyme–substrate binding between RasGAP and Ras.)

Taken from
Kurt W. Kohn

Kitano Diagrams

From direct graphical representation of chemical reactions

To more abstract representation of the logic such reactions implement



(a) Graphical representation of fission yeast Cdc2 in Kohn diagram

(b) Proposed improvements of graphical representation of fission yeast Cdc2

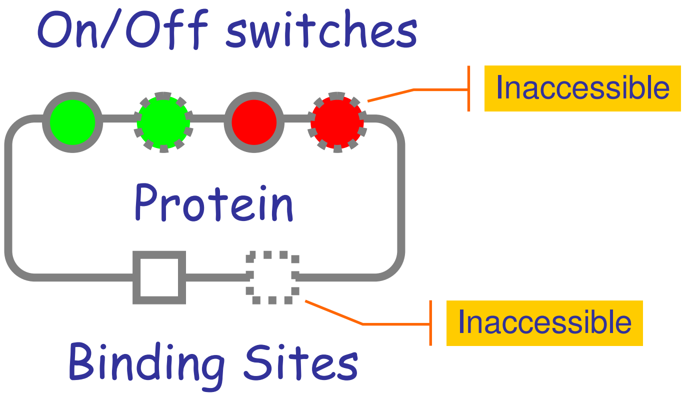
BioSilico

Figure 1. Representation of fission yeast Cdc2 protein in **(a)** the original MIM and **(b)** proposed improvements. Both diagrams represent interactions involving fission yeast Cdc2. Wee1 phosphorylates Thr14 and Tyr15, Mik1 phosphorylates Tyr15, Mcs6 phosphorylates Thr167, and Cdc25 dephosphorylates Thr14 and Tyr15. Cdc2 binds to either Cdc13, Cig1, or Cig2. When Cdc2 is forming a complex with Cdc13 and only Thr167 is phosphorylated, the complex interacts with Lamina. Phosphorylation of either Thr14 or Tyr15 inhibits activation of Cdc2 due to phosphorylation of Thr167. The complex auto-phosphorylates Try15 of its Cdc2. The complex of Cdc2 and Cig1 interacts with Rum1. Cdc2-Cdc13 complex and Cdc-Cig2 complex form heterotrimers involving Rum1.

Taken from Hiroaki Kitano

The “Instruction Set” of the Protein Machine

cf. BioCalculus [Kitano&Nagasaki], κ-calculus [Danos&Laneve]



On/Off switches

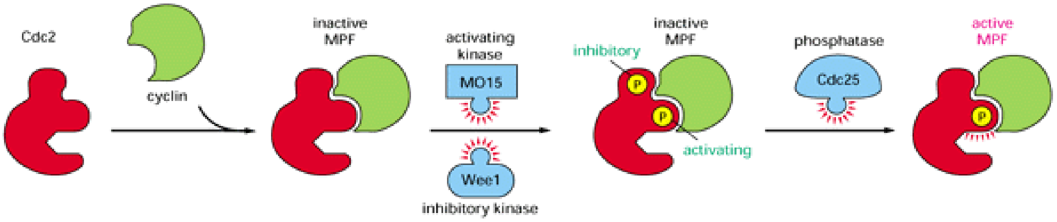
Inaccessible

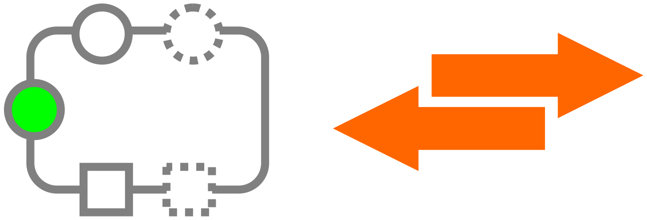
Protein

Inaccessible

Binding Sites

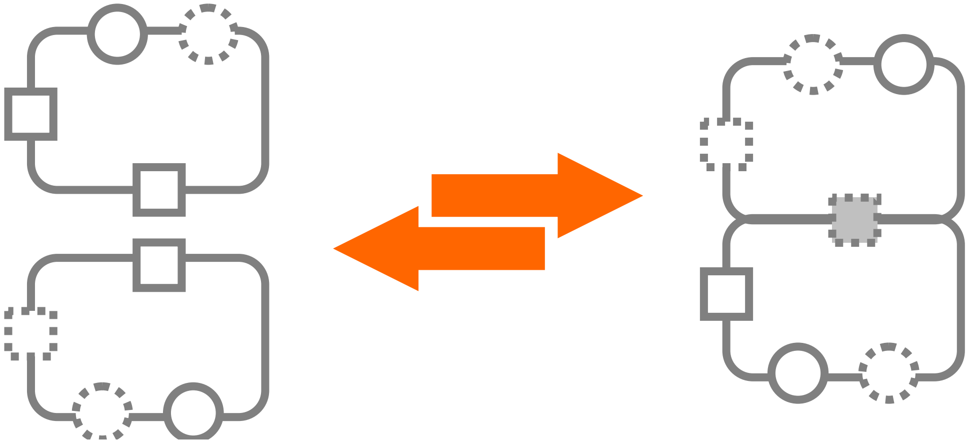
Each protein has a structure of binary switches and binding sites. But not all may be always *accessible*.





Switching of accessible switches.

- May cause other switches and binding sites to become (in)accessible.
- May be triggered or inhibited by nearby specific proteins in specific states.

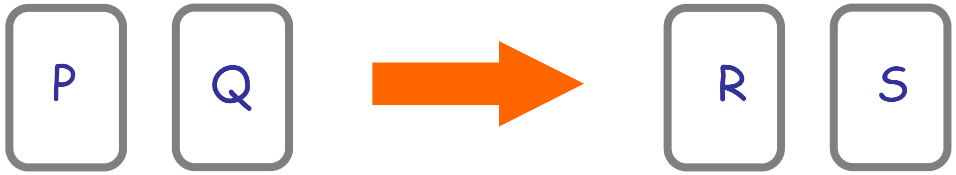


Binding on accessible sites.

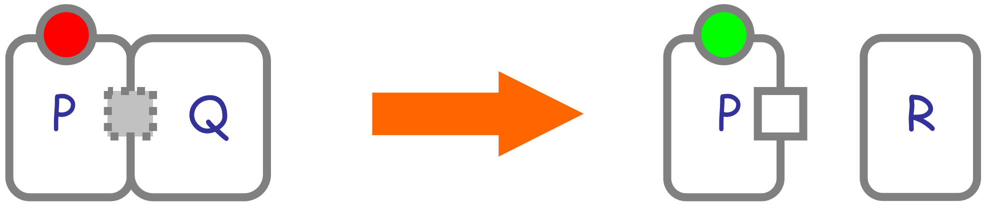
- May cause other switches and binding sites to become (in)accessible.

-- May be triggered or inhibited by nearby specific proteins in specific states.

…



Ordinary Chemical Reactions

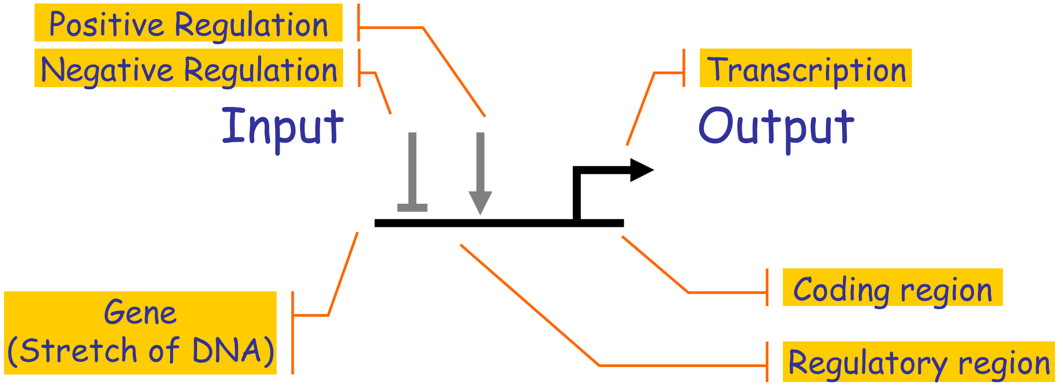


Any combination of the above

Abstractions of the Protein Machine

- BioSPi
  - Regev-Shapiro-Silverman propose modeling chemical interactions (exchange of electrons and small molecules) as “communication”.
  - Standard stochastic simulation algorithms (Gillespie) can be used to run in-silico experiments.
  - Complex formation is encoded via p-restriction.
- Stochastic p-Calculus
  - Priami formalizes a stochastic version of π-calculus where channels have communication *rates*.
- k-calculus
  - Danos and Laneve (following Kitano’s BioCalculus) define a calculus where complex formation is primitive.
- Bio State Charts
  - Harel uses State Charts to model biological interactions via a semi-graphical FSM notation.
- Pathway Logic
  - Talcott-Eker-Knapp-Lincoln use term-rewriting.
- BioCham
  - ChabrierRivier-Fages-Soliman use term-rewriting and CLT modelchecking.
- …
- SBML (Systems Biology Markup Language)
  - XML dialect for MIM’s:
    - Compartments (statically nested)
    - Reagents with concentrations
    - Reactions with various rate laws
  - Read and written by many tools via the Systems Biology Workbench protocol
    - Graph editors
    - Simulators (including simulation web services)
    - Databases

2. The Gene Machine



Regulation of a gene (positive and negative) influences transcription. The regulatory region has precise DNA sequences, but not meant for coding proteins: meant for binding regulators.

Transcription produces molecules (RNA or, through RNA, proteins) that bind to regulatory region of other genes (or that are end-products).

Human (and mammalian) Genome Size
3Gbp (Giga base pairs) 750MB @ 4bp/Byte (CD)
Non-repetitive: 1Gbp 250MB
In genes: 320Mbp 80MB
Coding: 160Mbp 40MB
Protein-coding genes: 30,000-40,000

M.Genitalium (smallest true organism)
580,073bp 145KB (eBook)
E.Coli (bacteria): 4Mbp 1MB (floppy)
Yeast (eukarya): 12Mbp 3MB (MP3 song)
Wheat 17Gbp 4.25GB (DVD)

A Gene Regulatory Network

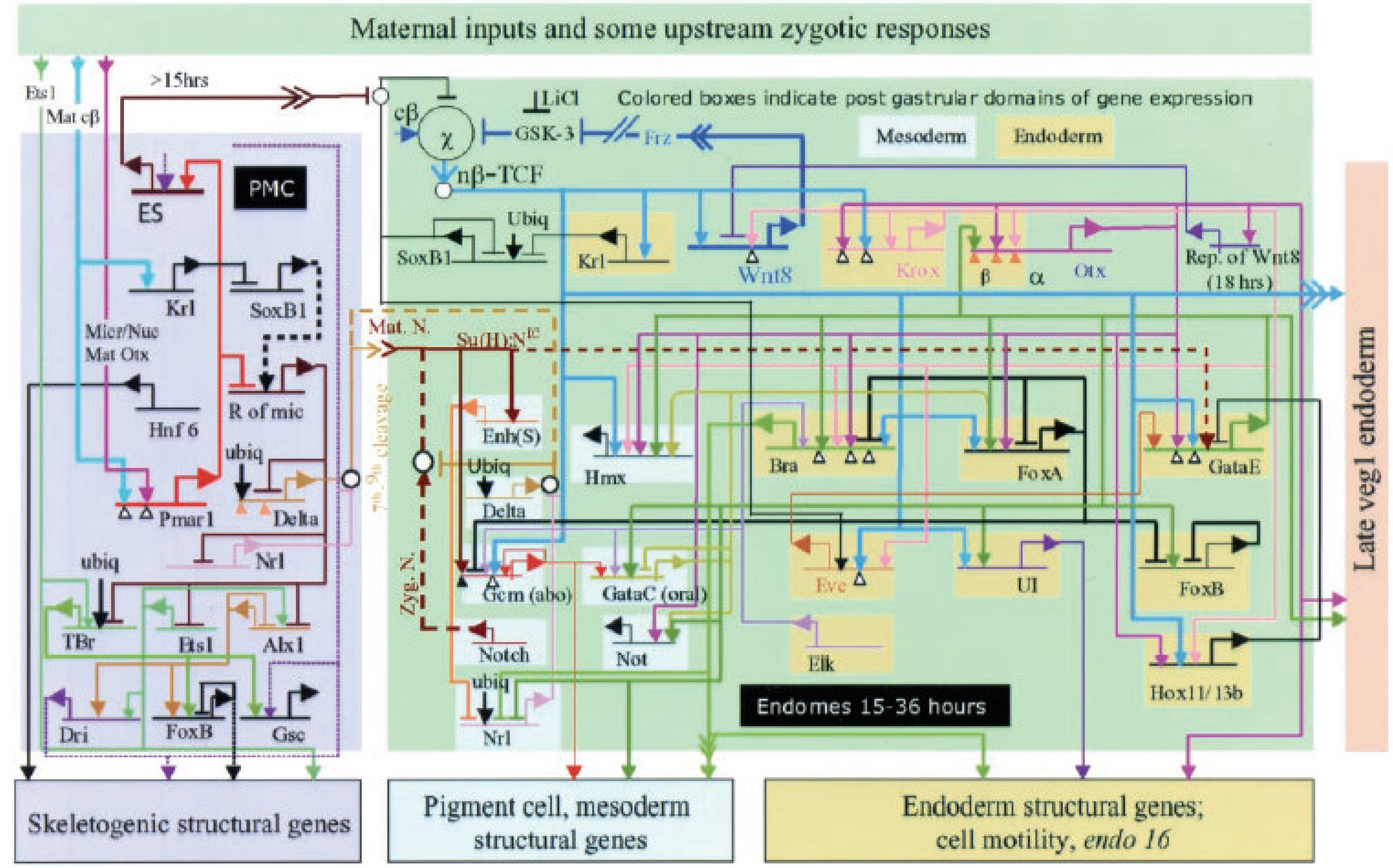
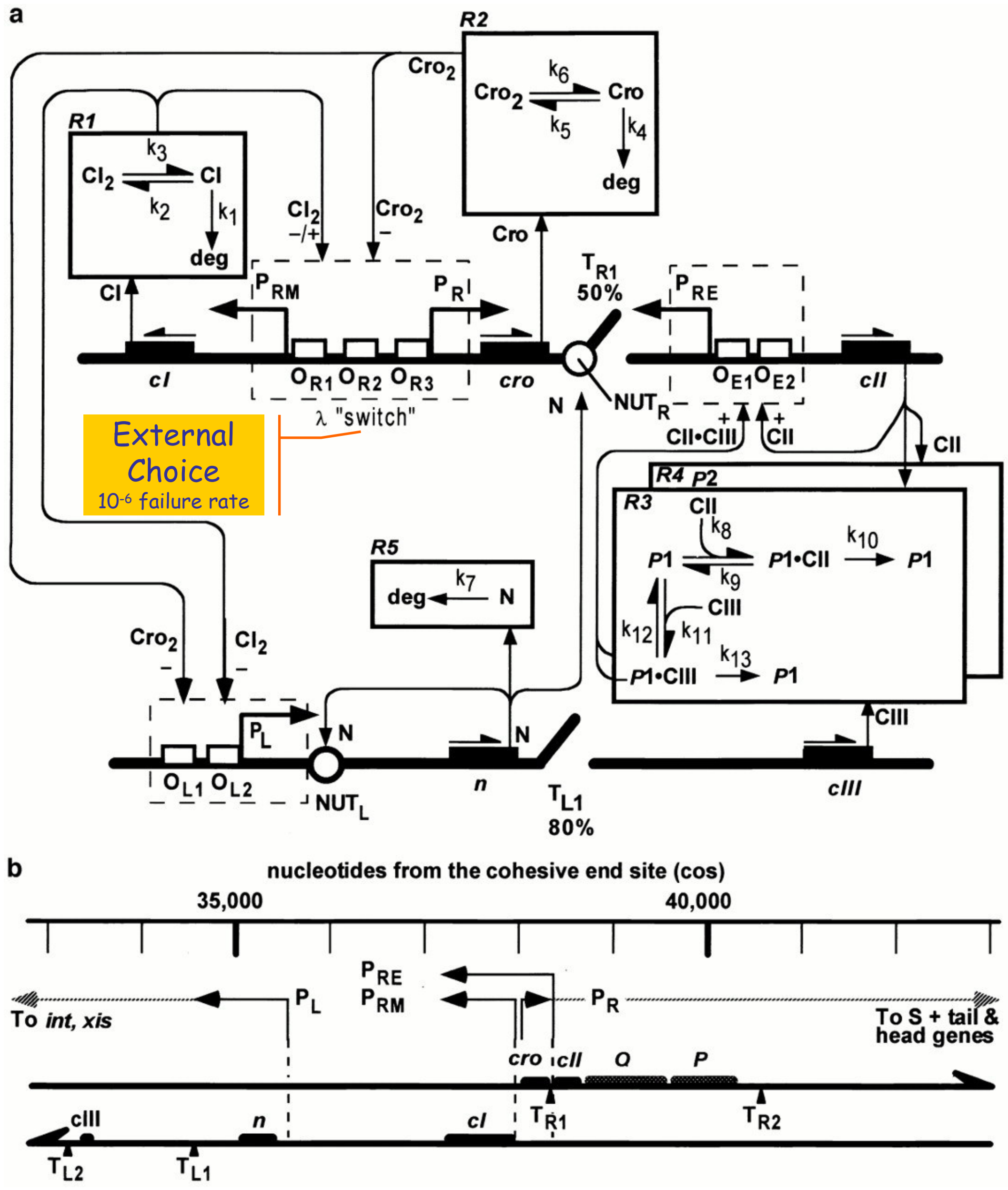


Fig. 1. Central portion of the *Strongylocentrotus purpuratus* embryo endomesoderm GRN, from fertilization to just before gastrulation. The diagram is a recent version of that initially presented in refs. 9–11. Suspected interactions at the cis-regulatory elements represented by the horizontal lines are shown, irrespective of when in the 0- to 30-h period or where in the embryo they are expected to occur [a "view from the genome" GRN (24); for interactions occurring only in given domains and at given periods see ref. 10 and www.its.caltech.edu/~mirsky/endomes.htm]. Transcriptional regulatory interactions are shown in the indicated spatial domains of the embryo: pmc domain, the skeletogenic micromere lineage; endomes domain, endomesoderm descendant from the sixth cleavage ring of eight "veg2" cells (2, 13, 24). Transcriptional inputs into the cis-regulatory elements of each named gene are indicated by arrows (activation, or permissive of activation) or bars (repression). Outputs from each gene (where known) are indicated by color-coded lines emanating from the bent arrows that symbolize transcription. For evidence see text, refs. 9–11, 15, 16, and 18, and www.its.caltech.edu/~mirsky/endomes.htm. An arrowhead inserted in an arrow tail indicates an intercellular signaling interaction; small open circles indicate cytoplasmic interactions or specific events off the DNA, e.g., that by which the Soxb1 factor interferes with nuclearization of β-catenin (26). For further details see refs. 9 and 10 and www.its.caltech.edu/~mirsky/endomes.htm.

Taken from
Eric H Davidson

Phage Lambda Decision Circuit



External Choice
10^{-6} failure rate

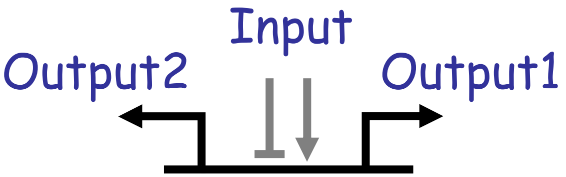
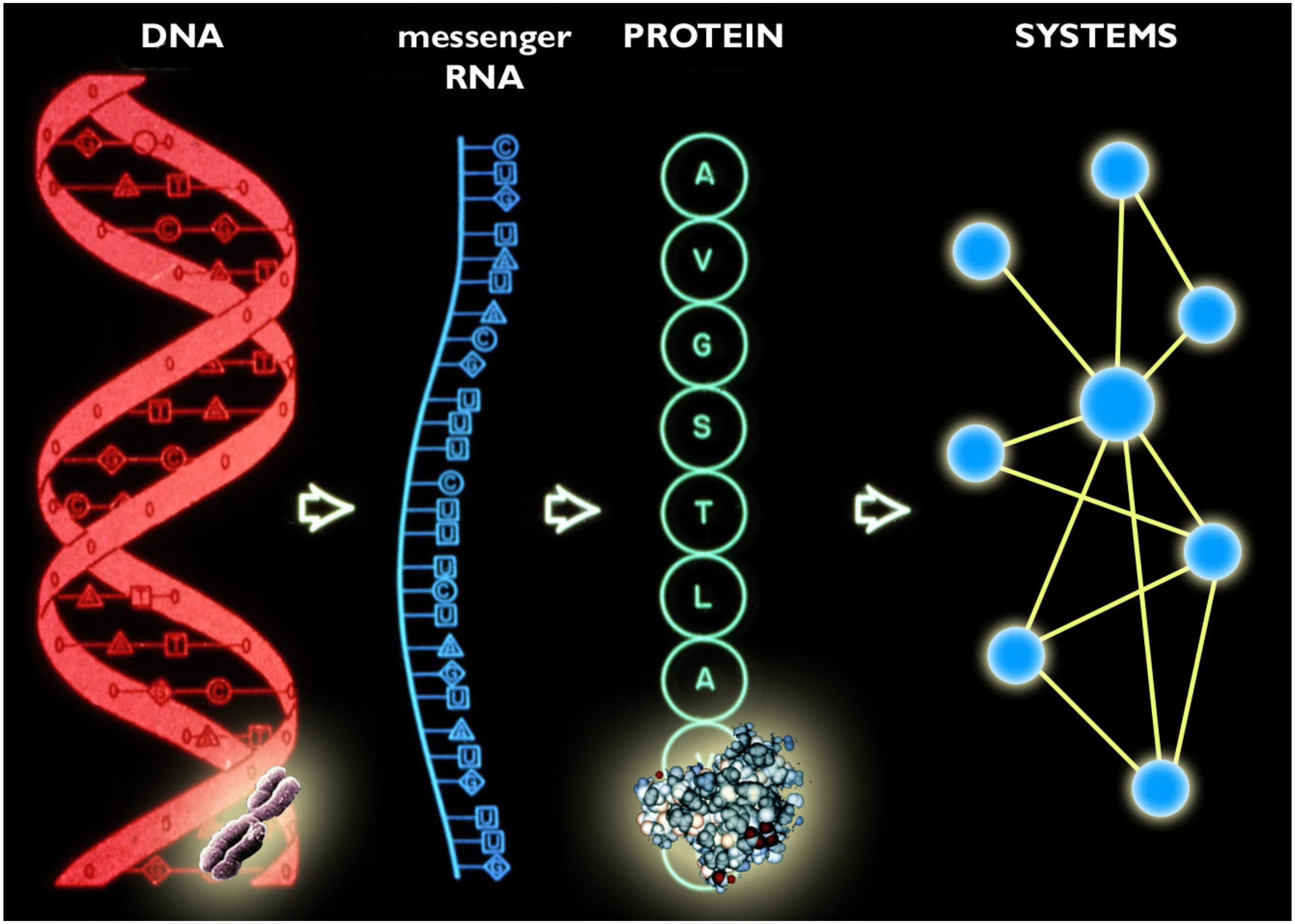


Figure 1. The phage lysis-lysogeny decision circuit. (a) Bold horizontal lines indicate stretches of double-stranded DNA. Arrows over genes indicate direction of transcription. Dashed boxes enclose operator sites that comprise a promoter control complex. The three operator sites, *O*R1–3, of the "lambda switch" implement concentration-dependent logic controlling promoters *P*RM and *P*R. Cro and CI dimers bind to the three sites with different affinities and in opposite order to control the activation level of the *P*RM and *P*R promoters (PTASHNE 1992 _; SHEA and ACKERS 1985 _). The five boxes R1–R5 contain nongenetic protein reaction subsystems. In *R*1, *R*2, and *R*5, "deg" indicates degradation. When protein N is available, transcribing RNAPs can be antiterminated at the NUTR and NUTL sites; termination sites *T*R1 and *T*L1 are inoperative for antiterminated RNAPs. The CI dimer acts as either a repressor or activator of promoter *P*RM, depending on its concentration. See text for discussion of the proteases labeled as *P*1 and *P*2 in *R*3 and *R*4. (b) decision circuit DNA organization. Phage-encoded genetic elements of the decision circuit are located in a 5000 nucleotide region of the phage DNA. Genes are separated onto leftward and rightward transcribed strands as indicated by the arrows. Rightward extensions of the antiterminated *P*R transcript transcribe the O and P genes essential for phage genome replication and the Q gene that controls transcription of later genes on the lytic pathway. Leftward extension of the antiterminated *P*L transcript transcribes *xis* and *int* genes essential for phage chromosome integration and excision into and out of the host chromosome. Locations of four termination sites are indicated by *T*R1–2 and *T*L1–2.

Taken from Adam Arkin

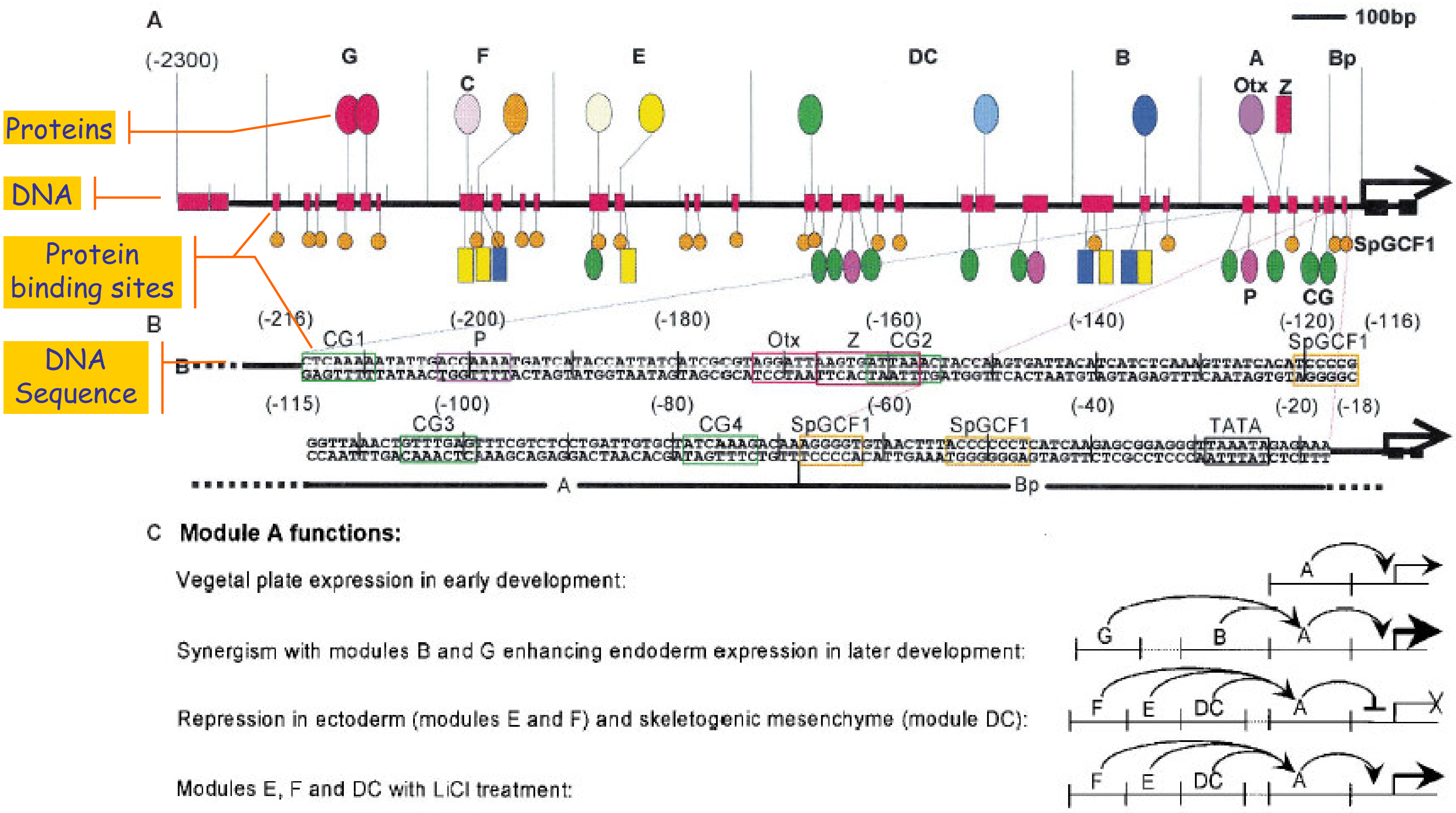
Structure of the Coding Region

- The Central Dogma of Molecular Biology:



Taken from Leroy Hood

Structure of a Regulatory Region


Proteins
DNA
Protein binding sites
DNA Sequence
A
100bp
(-2300)
G
F
C
E
DC
B
A
Otx
Z
Bp
SpGCF1
P
CG
B
(-216)
CG1
(-200)
P
(-180)
Otx
Z
(-160)
CG2
(-140)
(-120)
SpGCF1
(-116)
CTCAAAAAATATTGACCAAAATGATCATACCATTATCATCGCGTAGGATTAAGTGATTAAACTACCAAGTGATTACATCATCTCAAAGTTATCACATCCCCG
GAGTTTTTTATAACTGGTTTTACTAGTATGGTAATAGTAGCGCATCCTAATTCACTAATTTGATGGTTCACTAATGTAGTAGAGTTTCAATAGTGTAGGGGC
(-115)
(-100)
CG3
(-80)
CG4
SpGCF1
(-60)
SpGCF1
(-40)
TATA
(-20)
(-18)
GGTTAAACTGTTTGAGTTTCGTCTCCTGATTGTGCTATCAAAGACAAAGGGGTGTAACTTTACCCCCCTCATCAAGAGCGGAGGGTTAAATAGAGAAA
CCAATTTGACAAACTCAAAGCAGAGGACTAACACGATAGTTTCTGTTTCCCCACATTGAAATGGGGGGAGTAGTTCTCGCCTCCCAATTTATCTCTTT
A
Bp
C Module A functions:
Vegetal plate expression in early development:
Synergism with modules B and G enhancing endoderm expression in later development:
Repression in ectoderm (modules E and F) and skeletogenic mesenchyme (module DC):
Modules E, F and DC with LiCl treatment:
A
G
B
A
F
E
DC
A
F
E
DC
A

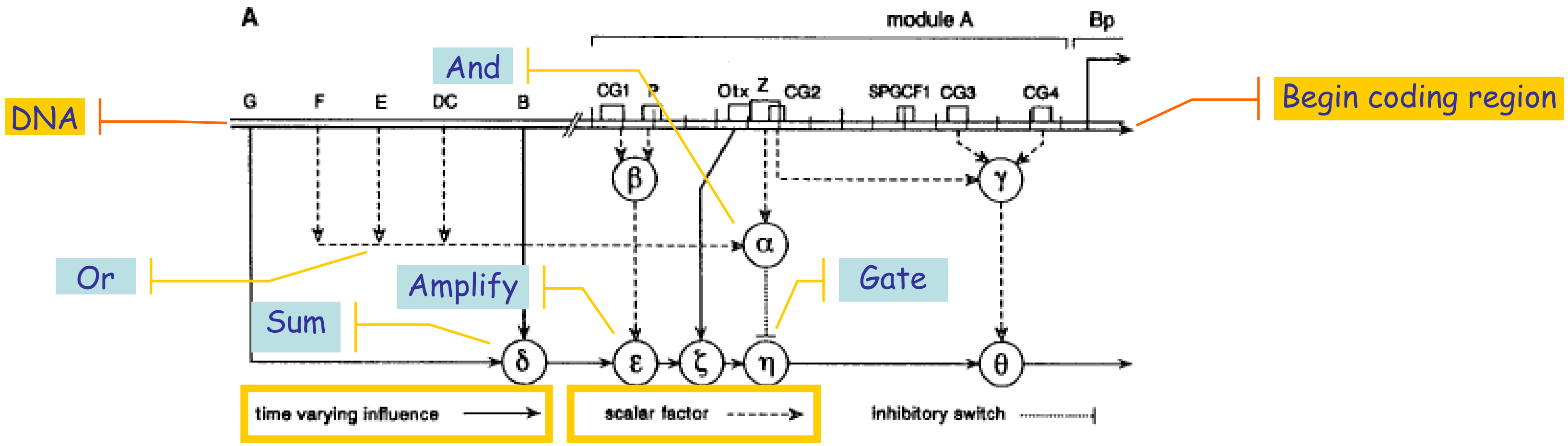
2300bp!

> average protein

Fig. 1. *Endo16* cis-regulatory system and interactive roles of module A. **(A)** Diversity of protein binding sites and organization into modular subregions [modified from (7)]. Specific DNA binding sites are indicated as red blocks; modular subregions are denoted by letters G to A (Bp, basal promoter). Proteins binding at the target sites considered in this work are indicated: Otx, SpOtx-1 (*12*); SpGCF1 (*14*); the proteins CG, Z, and P, which are not yet cloned; and protein C [a CREB family protein (*18*)] in subregion F. Proteins for which sites occur in multiple regions of the DNA sequence (indicated by the black line) are shown beneath. **(B)** Sequence of module A and location of protein binding sites. Sites are indicated in the same colors as in (A). A fragment containing CG_3 and CG_4 sites as well as Bp has no endoderm-specific activity and services other upstream cis-regulatory systems promiscuously; similarly, the *Endo16* cis-regulatory system functions specifically with heterologous promoters substituted for Bp (*5, 8, 19*). Boxed sequences indicate conserved core elements of the target sites (*7, 12, 14*), not the complete target site sequences. **(C)** Integrative and interactive functions of module A (*5, 8*). Module A communicates the output of all upstream modules to the basal transcription apparatus. It also initiates endoderm expression, increases the output of modules B and G, and is required for functions of the upstream modules F, E, and DC. These functions are repression of expression in nonendodermal domains and enhancement of expression in response to LiCl.

Taken from Eric H Davidson

Function of a Regulatory Region



And

DNA

Begin coding region

Or

Amplify

Gate

Sum

B

if (F = 1 or E = 1 or CD = 1) and (Z = 1)
$\alpha = 1$
else $\alpha = 0$
	Repression functions of modules F, E, and DC mediated by Z site

if (P = 1 and CG_1 = 1)
$\beta = 2$
else $\beta = 0$
	Both P and CG_1 needed for synergistic link with module B

if (CG_2 = 1 and CG_3 = 1 and CG_4 = 1)
$\gamma = 2$
else $\gamma = 1$
	Final step up of system output

$\delta(t) = B(t) + G(t)$	Positive input from modules B and G
$\varepsilon(t) = \beta^*\delta(t)$	Synergistic amplification of module B output by CG_1-P subsystem

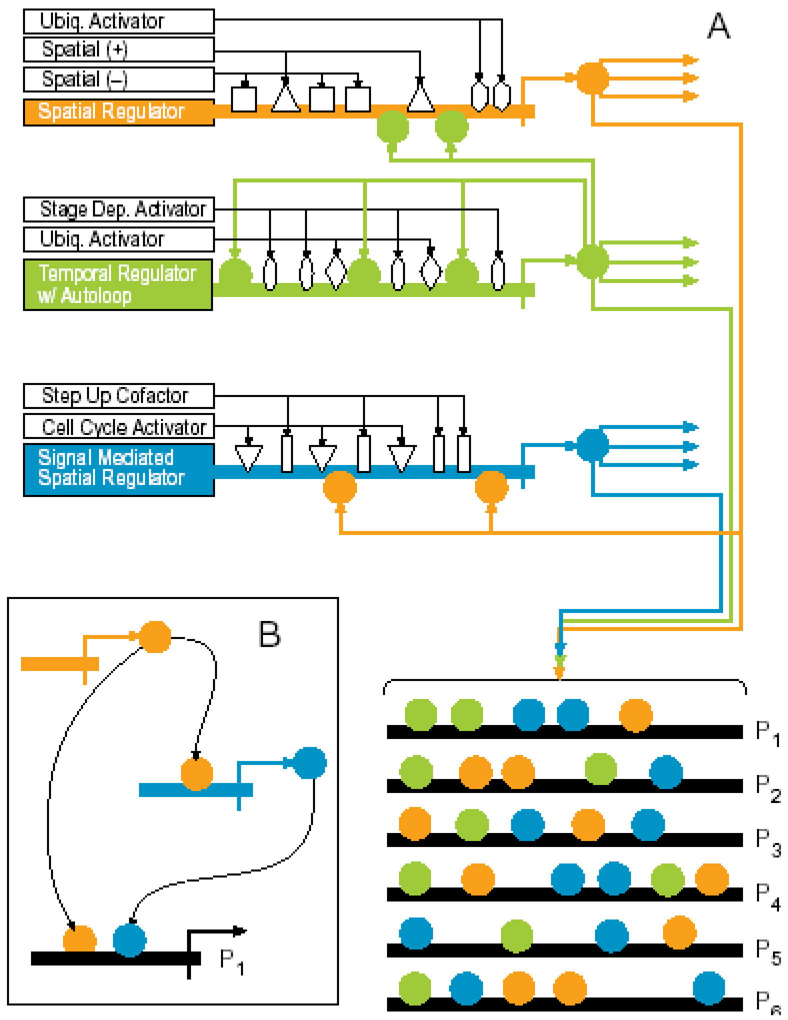
if ($\varepsilon(t) = 0$)
$\xi(t) = Otx(t)$
else $\xi(t) = \varepsilon(t)$
	Switch determining whether Otx site in module A, or upstream modules (i.e., mainly module B), will control level of activity

if ($\alpha = 1$)
$\eta(t) = 0$
else $\eta(t) = \xi(t)$
	Repression function inoperative in endoderm but blocks activity elsewhere

$\Theta(t) = \gamma^*\eta(t)$	Final output communicated to BTA

Taken from
Eric H Davidson

Where/When/HowMuch



3 genes encoding transcription factors

6 genes encoding proteins

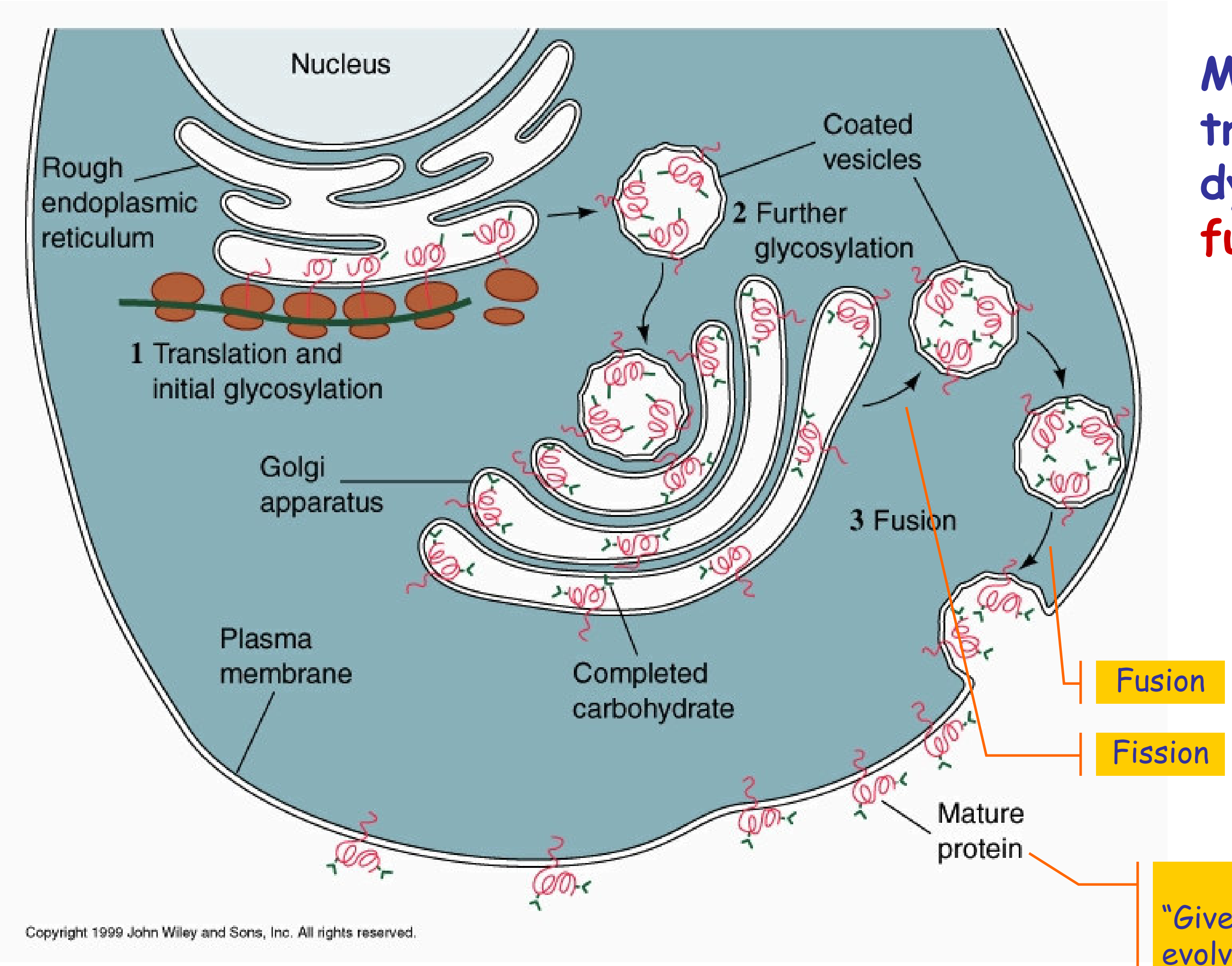
Taken from Eric H Davidson

Abstractions of the Gene Machine

- Hybrid Petri Nets
  - [Matsuno, Doi, Nagasaki, Miyano]
- …

- Many of the same techniques as for the Protein Machine apply.
  - Process Calculi
  - Term-Rewriting Systems
  - …

3. The Membrane Machine



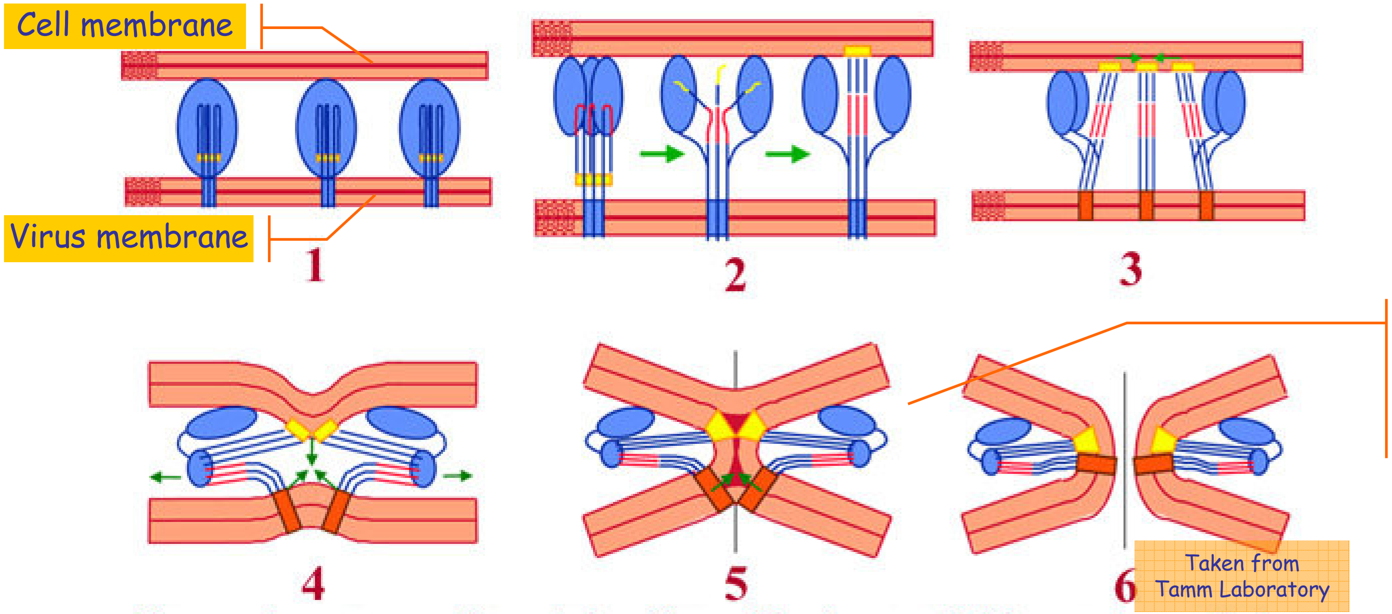
Molecular transport and transformation through dynamic compartment fusion and fission.

These "Life of a Saint" diagrams (all temporal stages shown at once) are popular because this is what people actually see in microscopes.

Well, what is all that for?
"Given the complicated pathways that have evolved to synthesize them, it seems likely that these [modified proteins] have important functions, but for the most part these functions are not known" [MBP p.609]

Membrane Fusion

Positive curvature to Negative curvature transition in 3D

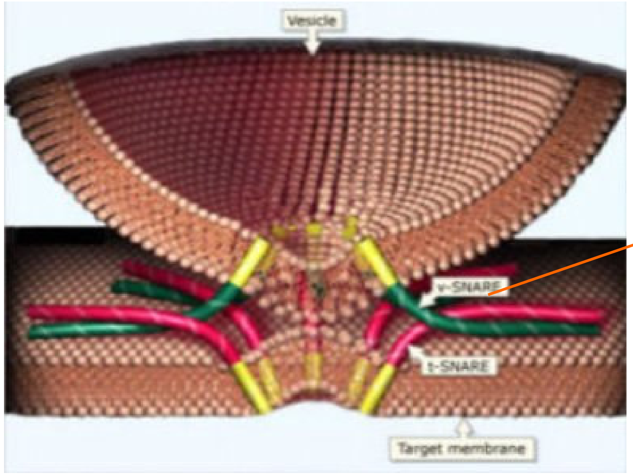


Taken from Tamm Laboratory

Proposed sequence of events in pH sensitive hemagglutinin membrane fusion

Aggressive fusion
(virus)

"By unknown mechanisms, the exoplasmic leaflets of the two membranes fuse" [MCB p745]

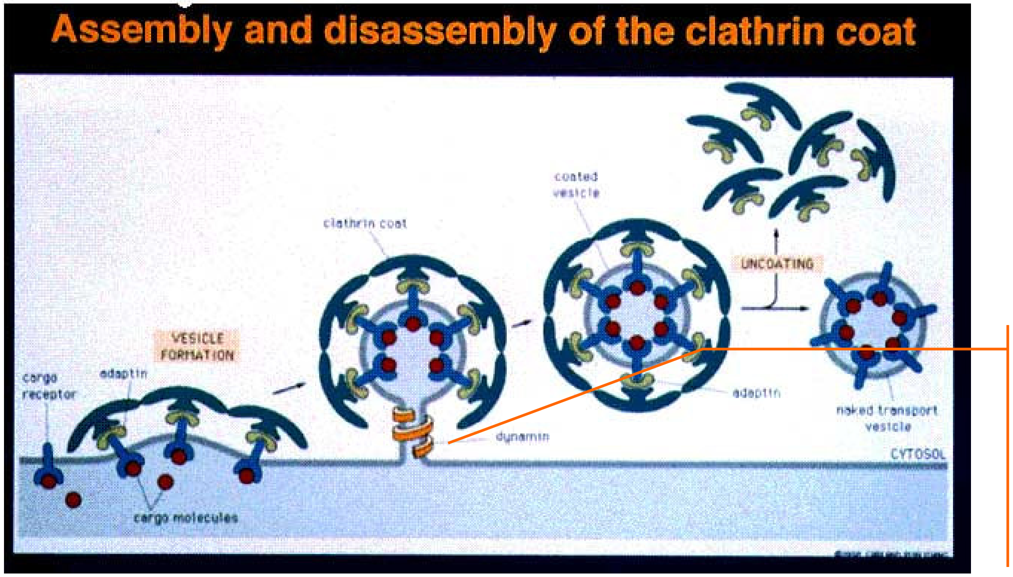


Cooperative fusion
(vesicle)

"Fusion of the two membranes immediately follows prefusion, but precisely how this occurs is not known" [MCB p742]

Membrane Fission

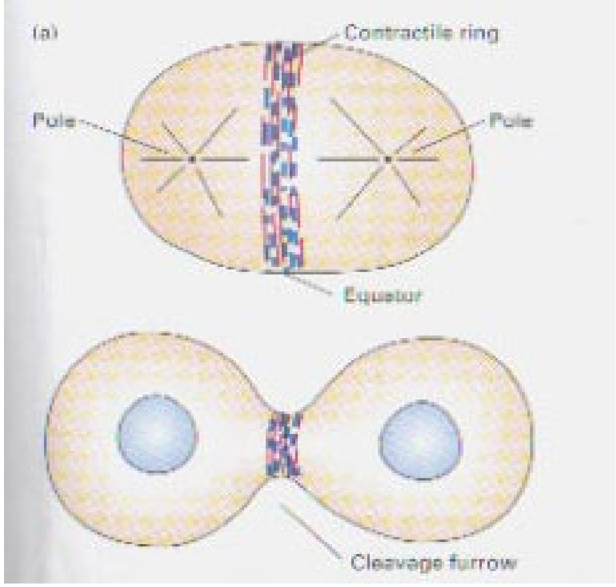
Negative curvature
to
Positive curvature
transition in 3D


Assembly and disassembly of the clathrin coat

Vesicle Formation

Movie by Allison Bruce

"Nonetheless, the actual process whereby a segment of phospholipid bilayer is 'pinched off' to form a pit and eventually a new vesicle is still not understood" [MCB p.746]

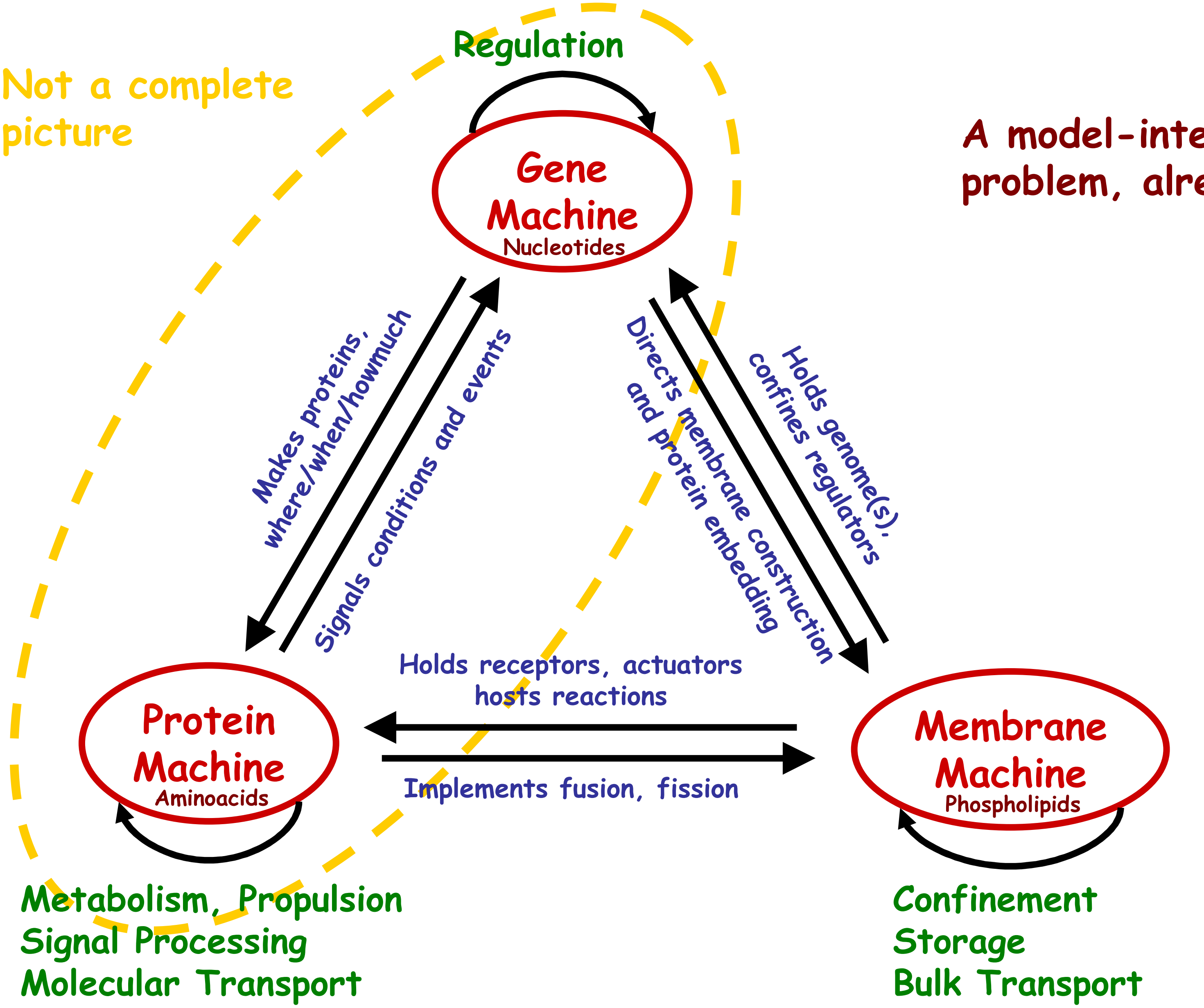


Cytokinesis (Mitosis)

Abstractions of the Membrane Machine

- P-Systems
  - G.Paun uses ideas from the theory of grammars and formal languages to model "Membrane Computing" (book 2002).
  - Some aspects not a good match from a "process" point of view (notions of termination, lock-step execution, static compartments), but field is evolving: http://psystems.disco.unimib.it/.
- BioAmbients
  - An extension of BioSPi along Ambient Calculus lines (with more bio-relevant mobility primitives) to model dynamic compartments.
- Brane Calculi
  - Computation *on* the membrane...

4. Summary



Not a complete picture
Regulation
Gene Machine
Nucleotides
Makes proteins, where/when/howmuch
Signals conditions and events
Directs membrane construction and protein embedding
Holds genome(s), confines regulators
Protein Machine
Aminoacids
Holds receptors, actuators hosts reactions
Implements fusion, fission
Membrane Machine
Phospholipids
Metabolism, Propulsion
Signal Processing
Molecular Transport
Confinement
Storage
Bulk Transport

A model-integration problem, already

Modeling Stuff with Process Calculi

In Their Own Words…

- On the nature of modeling
  - Sydney Brenner: *"When you want to have a predictive science, you have to be able to calculate."*
  - Hamid Bolouri & Eric H. Davidson: *"Abstract models have relatively few parameters and so … it is easier to explore their behavior and build models with them. … In contrast, more detailed models suffer from an explosion in the number of their parameters."*
  - Denis Noble: *"There will probably therefore be no unique model that does everything at all levels. … One of the first questions to ask of a model therefore is what questions does it answer best."*
  - Hiroaki Kitano: *"Molecular biology has uncovered a multitude of biological facts … but this alone is not sufficient for interpreting biological systems. … A system-level understanding should be the prime goal of biology."*
  - Al Hershey: *"Influential ideas are always simple. Since natural phenomena need not be simple, we master them, if at all, by formulating simple ideas and exploring their limitations."*

Write Things Down!

- When you want to calculate, you have to be able to write things down:
  - **Write down biological systems as *programs*, as if they were software systems**
    - Software is a *precise* (yet not quite *predictable*) notation for systems of high structural and combinatorial complexity.
    - Small programs can express highly complex behavior. Especially true in concurrency vs. deterministic chaos.
    - We don't use differential equations to write operating systems.
  - **Write them as *text* (not graphs), to better describe dynamic behavior**
    - Concurrency, nondeterminism, stochasticity.
    - Representing *processes*, not just data.
- *How* shall we write them down?
  - Need to choose a ***syntax***
    - Always a food fight.
    - But needed for tools to work on: simulation, analysis, storage, search.
  - In C++, Haskell, Prolog?
    - Not likely... We need highly concurrent analyzable formal languages.

Process Calculi

- Chemistry is ok
  - Yes, chemical reactions *are* a process calculus! In fact, chemical analogies inspired the early definitions of process calculi.
  - But a large biochemical system becomes a flat list of a huge number of reactions: modules and higher-level functional abstractions are lost in the soup.

- Process calculi are:
  - The modular representation of discrete concurrent processes.
- They are language-oriented
  - In order to be compositional.
    - Combining separate modules or systems should be easy.
  - To fully represent dynamics.
    - Process evolution should be implicit in process syntax.
  - Graphs don't usually cut it.
    - Either property above can fail in graph-oriented descriptions of processes.
    - Hence, process calculi do not usually make nice pictures.

Process Calculi

- Unfortunately, there are *many* process calculi.
  - There are suitable general theories (Milner's BiGraphs) where we can hope to achieve at least partial *model integration*.
  - π-calculus is "canonical": has most interesting things in it (composition, interaction, and hiding), but not all.
  - There is a set of standard techniques (transition systems, equivalences, etc.) to build and study new calculi "on demand".

- Fortunately, there are *many* process calculi.
  - Some are better for modeling software or hardware.
  - We can look for the ones that best model biological processes.
  - Many kinds of processes = many kinds of calculi.

Composition(ality)	$P \mid Q$
Complexation (new!)	$P{:}Q$
Localization	$[P]$
Interaction	$a.P$
Hiding	$(\nu n)P$

What Process Calculi Do For Us

- **We can write things down**
  - We can modularly describe high structural and combinatorial complexity ("do programming").
  - Software teaches us that large and deep systems, even well engineered ones where each component is rigidly defined, eventually exhibit "emergent behavior" (damn!).
- **We can calculate and analyze**
  - Directly support simulation.
  - Support analysis (e.g. control flow, causality, nondeterminism).
  - Support state exploration (modelchecking).
    - This was invented to discover "emergent behavior" (=bugs) in software and hardware systems.
    - Should have interesting large-scale applications in biology.
- **We can reason**
  - Suitable equivalences on processes induce algebraic laws.
  - We can relate different systems (e.g. equivalent behaviors).
  - We can relate different abstraction levels.
  - We can use equivalences for state minimization (symmetries).
- Disclaimers
  - Some of these technologies are basically ready (small-scale stochastic simulation and analysis, large-scale nondeterministic and stochastic modelchecking).
  - Others need to scale up significantly to be really useful. This is (has been) the challenge for computer scientists.
  - We don't use process calculi *either* to write operating systems, but we are working on that… the biggest and growing problems there are the management of concurrency, and the analyzability of software.