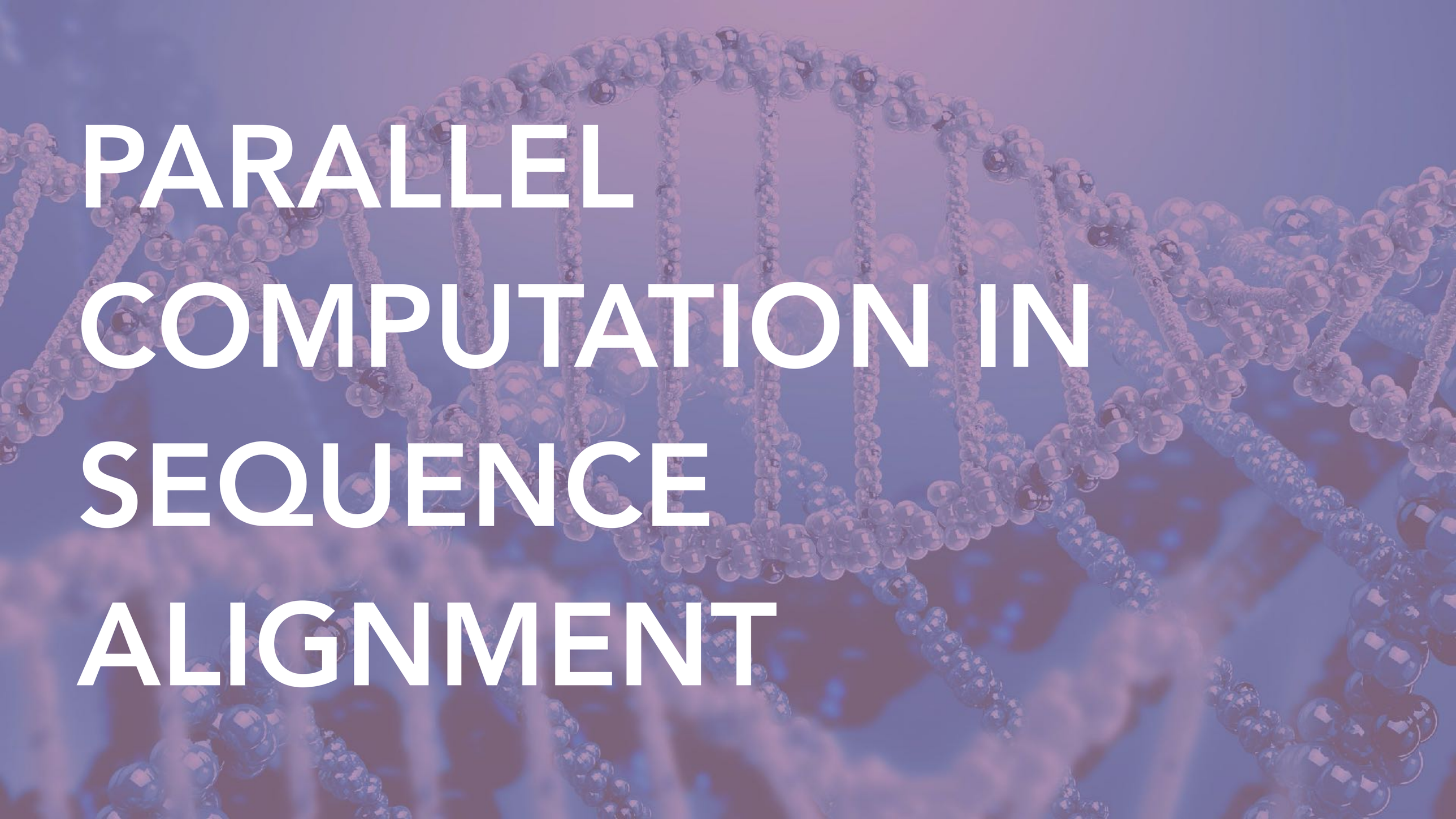
SESSION 6

SESSION 5

- High Performance Computing
- Protein Function, Modeling and Alignment



PARALLEL COMPUTATION IN SEQUENCE ALIGNMENT

PARALLEL COMPUTATION IN SEQUENCE ALIGNMENT

- Scanning and analyzing biological sequences are common and repeated tasks in molecular biology
  - Homologous sequence searching
    - Based on pairwise alignment
    - Task is to find similarities between a particular query sequence and all sequences in a database
  - Multiple sequence alignment
    - Simultaneous alignment of three or more nucleotide or amino acid sequences

Parallel Computation in Biological Sequence Analysis

Tieng K. Yap, Ophir Frieder, *Senior Member, IEEE*,
and Robert L. Martino, *Member, IEEE*

Abstract—A massive volume of biological sequence data is available in over 36 different databases worldwide, including the sequence data generated by the Human Genome project. These databases, which also contain biological and bibliographical information, are growing at an exponential rate. Consequently, the computational demands needed to explore and analyze the da contained in these databases is quickly becoming a great concern. To meet these demands, we must use high performance computing systems, such as parallel computers and distributed networks of workstations. We present two parallel computational methods for analyzing these biological sequences. The first method is used to retrieve sequences that are homologous to a query sequence. The biological information associated with the homologous sequences found in the database may provide important clues to the structure and function of the query sequence. The second method, which helps in the prediction of the function, structure, and evolutionary history of biological sequences, is used to align a number of homologous sequences with each other. These two parallel computational methods were implemented and evaluated on an Intel iPSC/860 parallel computer. The resulting performance demonstrates that parallel computational methods can significantly reduce the computational time needed to analyze the sequences contained in large databases.

ndex Terms—Sequence, comparison, alignment, search, retrieval, database, algorithm, parallel, speculative, computation.

TRODUCTION

E field of molecular biology has created many spe-
alized databases which contain diverse information,
as annotated biological sequences, three-dimensional
ular structures, and both genetic and physical maps.
et al. [31] have compiled a list of molecular biology
ases (LiMB database) which presently contains 189
s and is continuing to grow. The LiMB database listed
uence databases, including the internationally well-
h DNA sequence databases GenBank [5], EMBL [17],
DBJ [15], and the protein sequence databases PIR
WISS-PROT [1], and PDB [7].
ven the great deal of knowledge that can be derived
analyzing biological sequences, we developed parallel
ds to reduce the time required to perform two com-
onally intensive analyses: homologous sequence
ing and multiple sequence alignment. Our parallel
ing method reduces the retrieval time by nearly a

tein sequences, it took about 29 days on a single pr
but only about 13 hours on a 64 processor systen
parallel computational methods are also highly
That is, they can be implemented on either a sm
large number of processors. They can also be imple
on either a parallel computer or a network of works

Retrieving homologous sequences from existing d
is important to the biomedical research community
scientists discover new sequences, they are eager t
these databases for sequences that are similar or re
these discoveries. The biological information associa
the similar sequences found in the databases may
important clues for determining the structure and fu
the newly discovered ones. The detail similarity pa
the retrieved sequences can be seen in a multiple s
alignment, which is obtained by stacking up seque
top of each other. A minimal number of gaps are int

PARALLEL COMPUTATION IN SEQUENCE ALIGNMENT

- Problems with sequential solution
  - With the exponential growth of the sequence banks, homologous sequence searching becomes time consuming
  - The automatic generation of an accurate multiple alignment is computationally expensive
- Parallel solutions
  - Reduce computation time
  - Provide more accurate results
    - Less heuristics/approximation needed
    - Robust statistics

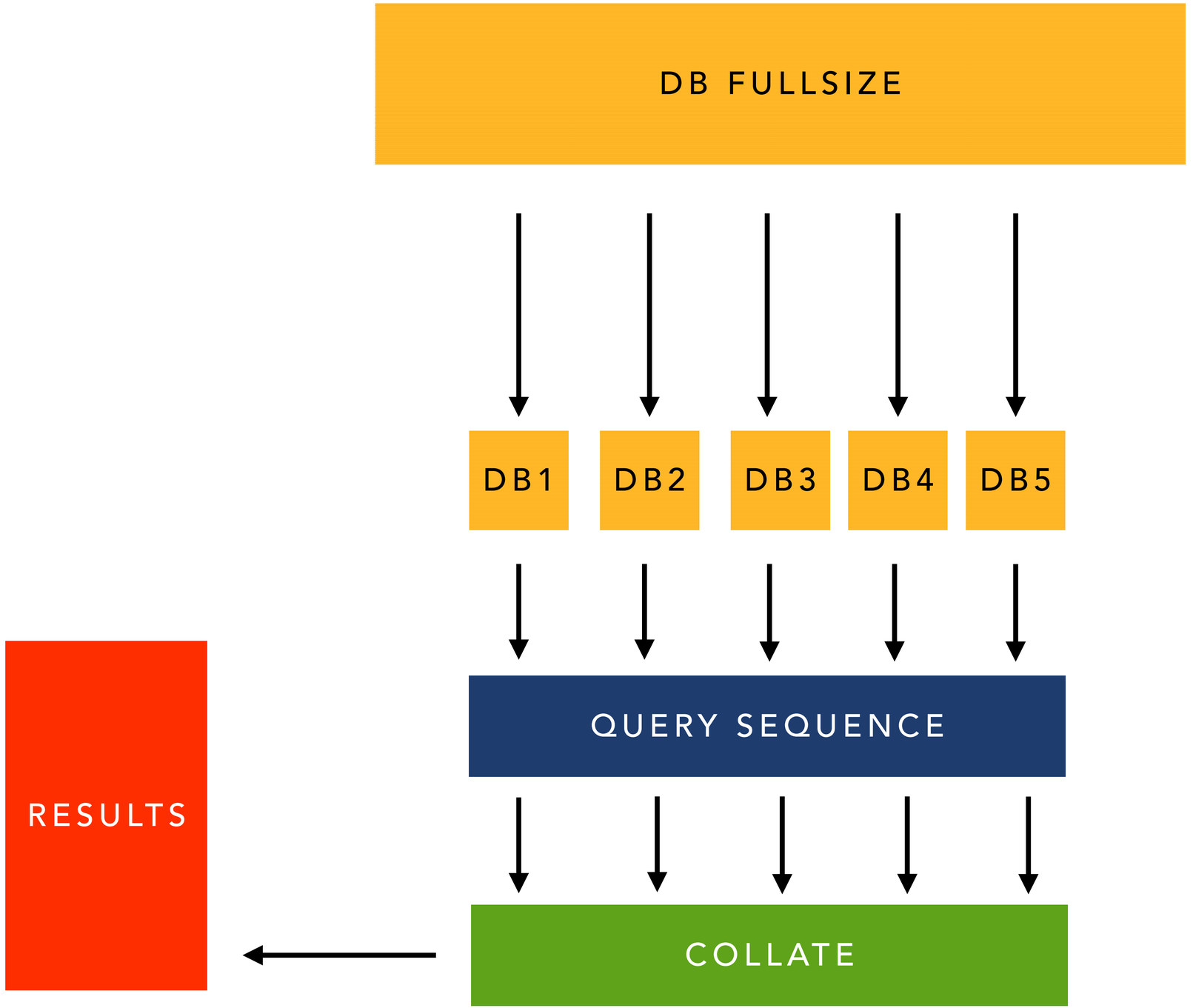
PARALLEL COMPUTATION IN SEQUENCE ALIGNMENT

- Parallel methods to search
  - Fine grain approach for SIMD parallel computer
    - Parallelize the comparison algorithm itself
    - All processors cooperate to determine the similarity score
  - Coarser grain approach for MIMD parallel computer
    - Parallelize the database searching
    - Each processor performs a selected number of comparison

FINE VS. COURSE

PARALLEL COMPUTATION IN SEQUENCE ALIGNMENT

- Parallelize Database Searching (coarse grain)
  - Partition database
  - Query against each partition
  - Combine results



PARALLEL COMPUTATION IN SEQUENCE ALIGNMENT

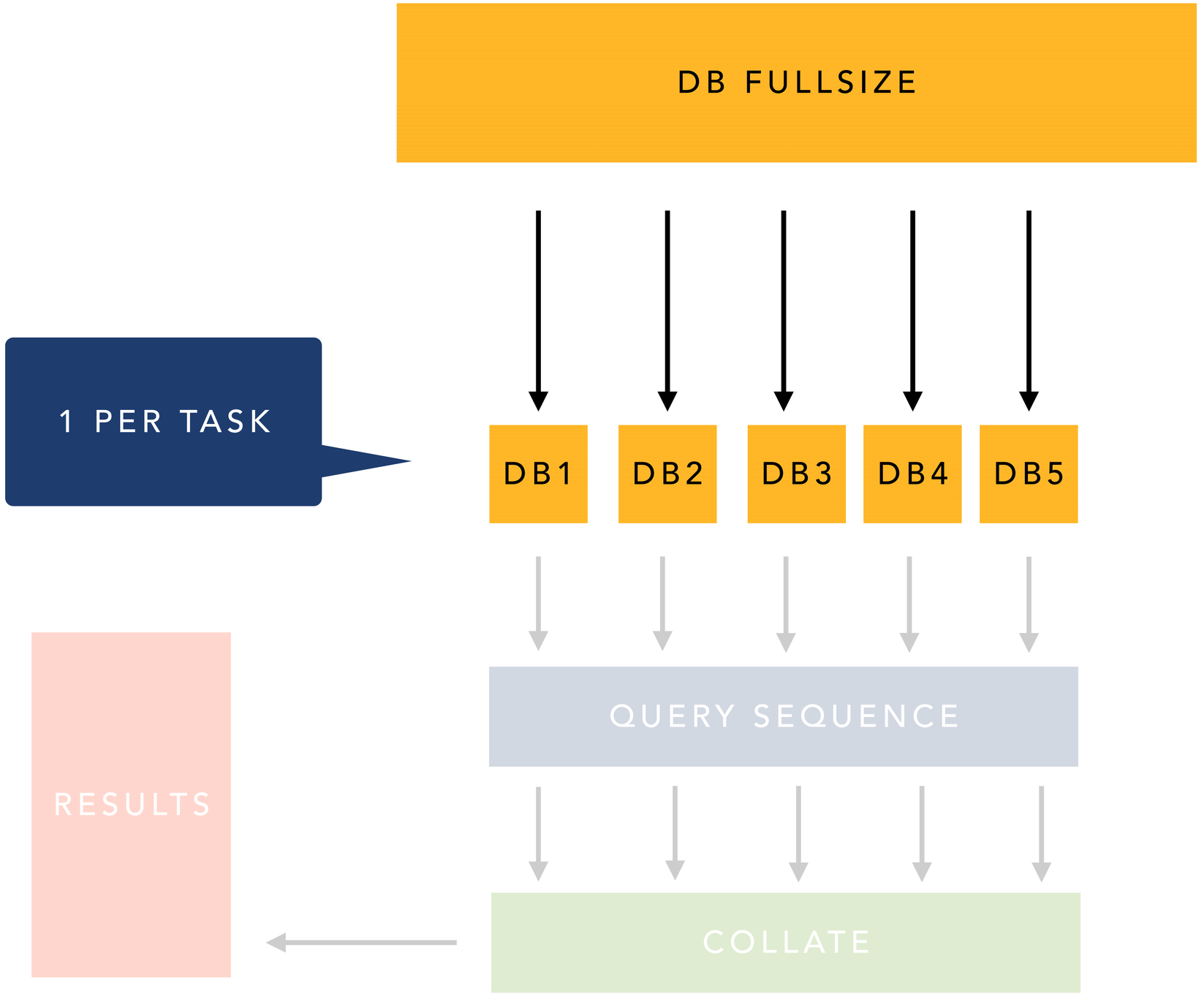
SPLITTING UP DATABASE

- Unsorted portion method

- Portion_size = database_size / processors_number

- Preprocessing

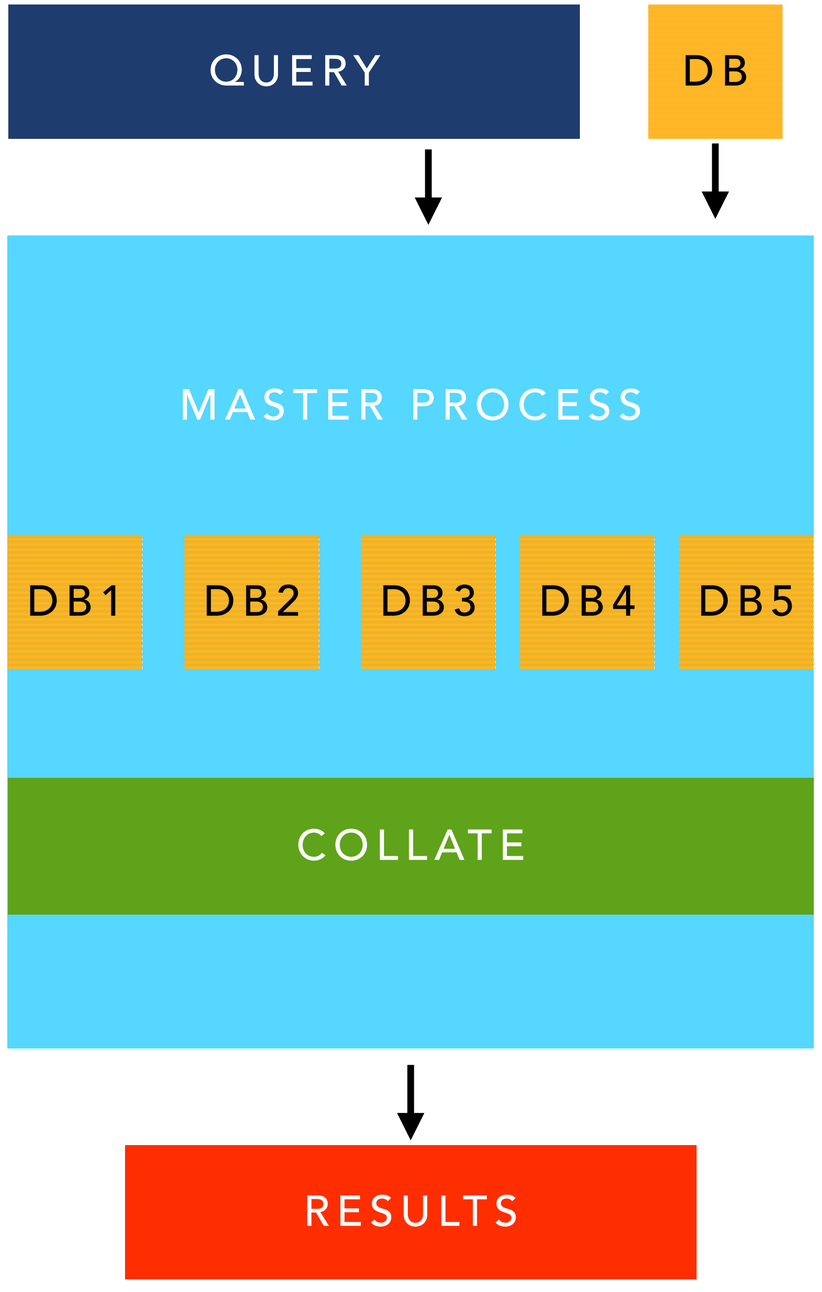
- Assumes a homogeneous cluster



PARALLEL COMPUTATION IN SEQUENCE ALIGNMENT

SPLITTING UP A DATABASE

- Sorted portion method – Master-worker method
  - Sequences are sorted in decreasing length order
  - The master processor distributes the sequences to the worker processors dynamically
    - Some nodes may be faster



MPI BLAST

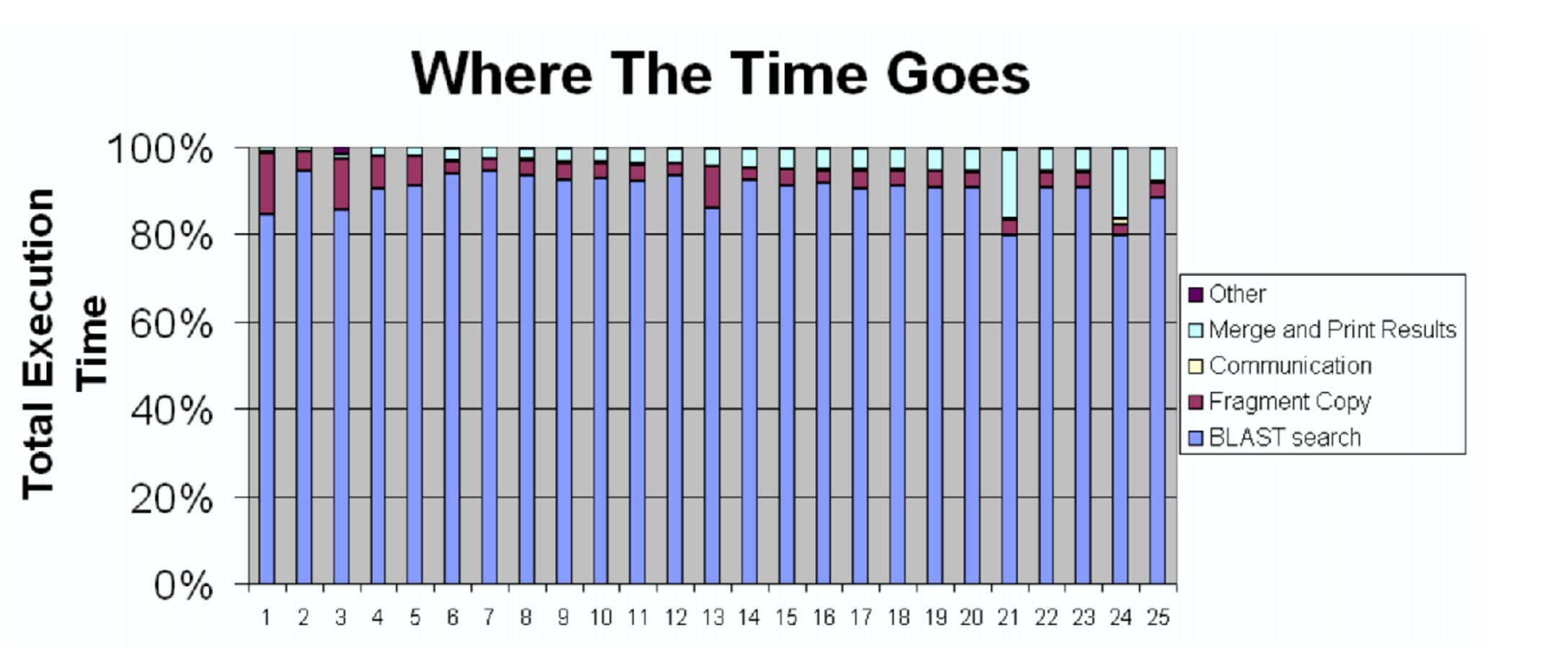
MPI BLAST

- mpiBlast is a tool to parallelizes the NCBI BLAST toolkit
  - Uses thread natively in execution
- mpiBLAST algorithm
  - Step 1: Database is segmented and placed on a shared storage device
  - Step 2: Queries are run on each node
    - If a node does not yet have a database fragment to search, it copies a fragment from shared storage
    - Fragment assignments to each node are determined by an algorithm that minimizes the number of fragment copies during each search

Algorithm 1 mpiBLAST master

Let *results* be the current set of BLAST results
Let $\mathbf{F} = \{f_1, f_2, ...\}$ be the set of database fragments
Let $\mathbf{Unsearched} \subseteq \mathbf{F}$ be the set of unsearched database fragments
Let $\mathbf{Unassigned} \subseteq \mathbf{F}$ be the set of unassigned database fragments
Let $\mathbf{W} = \{w_1, w_2, ...\}$ be the set of participating workers
Let $\mathbf{D}_i \subseteq \mathbf{W}$ be the set of workers that have fragment f_i on local storage
Let $\mathbf{Distributed} = \{\mathbf{D}_1, \mathbf{D}_2, ...\}$ be the set of $\mathbf{D}$ for each fragment
Require: $|\mathbf{W}| \neq 0$
Ensure: $|\mathbf{Unsearched}| = 0$
$\mathbf{Unsearched} \leftarrow \mathbf{F}$
$\mathbf{Unassigned} \leftarrow \mathbf{F}$
$results \leftarrow \emptyset$
Broadcast queries to workers
while $|\mathbf{Unsearched}| \neq 0$ **do**
  Receive a *message* from a worker w_j
  if *message* is a state request **then**
    if $|\mathbf{Unassigned}| = 0$ **then**
      Send worker w_j the state $SEARCH\_COMPLETE$
    else
      Send worker w_j the state $SEARCH\_FRAGMENT$
    end if
  else if *message* is a fragment request **then**
    Find f_i such that $\min_{\mathbf{D}_i \in \mathbf{Distributed}} |\mathbf{D}_i|$ and $f_i \in \mathbf{Unassigned}$
    if $|\mathbf{D}_i| = 0$ **then**
      Add w_j to $\mathbf{D}_i$
    end if
    Remove f_i from **Unassigned**
    Send fragment assignment f_i to worker w_j
  else if *message* is a set of search results for fragment f_i **then**
    Merge *message* with *results*
    Remove f_i from **Unsearched**
  end if
end while
Print *results*

MPI BLAST



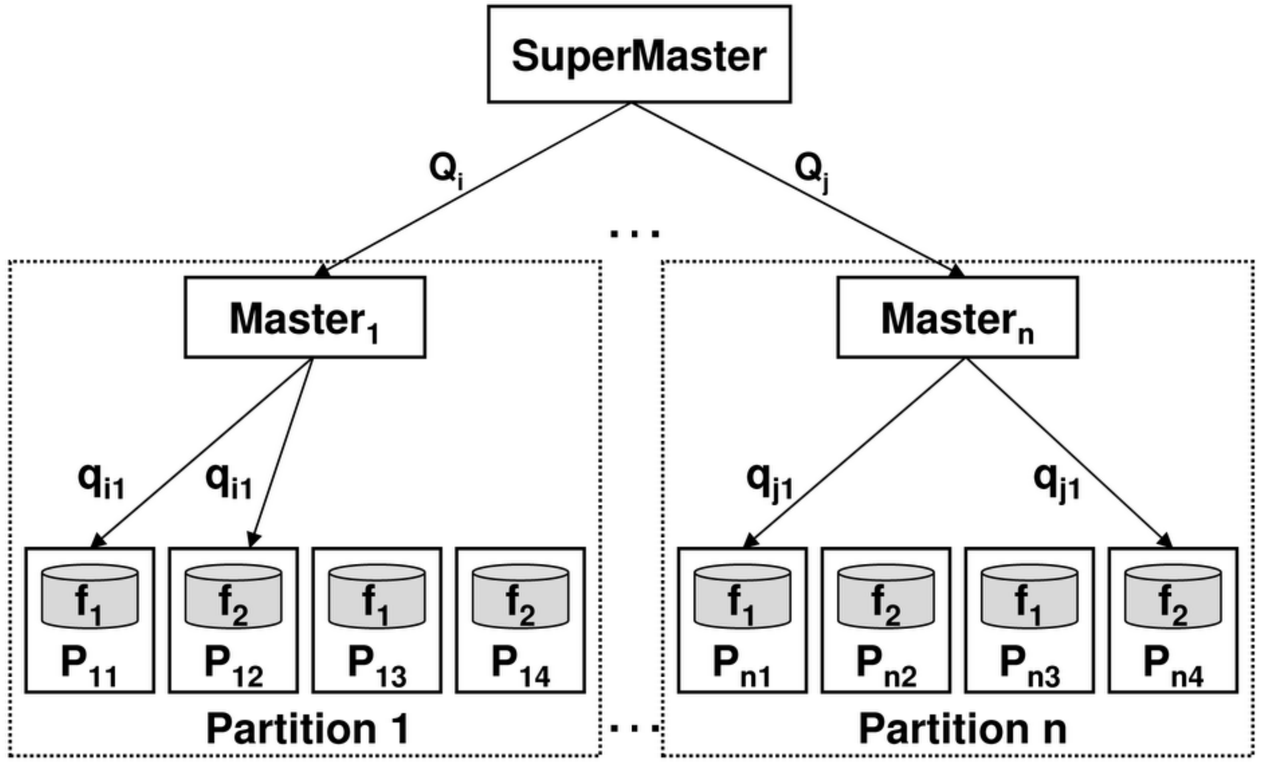
- How time is spend on worker nodes

MPI BLAST

- The "original" mpiBlast (ver.1.4 or older) design follows a database segmentation approach and a master-slave style
  - The master uses a greedy algorithm to assign pre-partitioned database fragments to workers
  - Each worker then concurrently performs a BLAST search on its assigned database fragment
  - The results from different workers are merged and written to the file system on the master
- The original mpiblast achieves good speedups when the number of processes is small or moderate
  - Scalability hampered by its centralized output processing design

MPI BLAST

- mpiBlast (1.6+) adopted a hierarchical architecture as
  - Processors in the system are organized into equally sized partitions which are supervised by a dedicated supermaster process
  - The supermaster process will handle interpartition load balancing
  - Within each partition there will be one master process and many worker processes
  - The master process is responsible for coordinating IO and computation scheduling in that partition



MPI BLAST

- Example command
  - mpirun -n 137 mpiblast --partition-size=17 --replica-group-size=2 --use-virtual-frags --use-parallel-write -p blastp -d testdb.fasta -i NC_015145.faa -o test1010/test.out --time-profile=test1010/tprofile.log -b 1 -m 8
- Explanations for options:
  - --partition-size will enable the hierarchical structure with multiple masters. If this option not used, mpiblast will run in the original way
    - partition-size = (num of workers) + (1 master process), and in the above case it's configured to have 16 workers within one partition
  - --replica-group-size specifies how db fragments will be replicated within one partition
    - In the above case, my db is pre-fragmented into 8 fragments, and I have 8 workers in one partition; so the replica-group-size=2 means the fragments are distributed on every 2 workers
  - --use-virtual-frags in this mode the db fragments are replicated to the worker nodes' memory. Since CCV cluster is a diskless platform, this option should be used
  - -n tells the system how many processes to run
    - In the above case, it should be calculated as ((num of workers) + (1 master process)) * (num of partitions) + (1 supermaster process) = (16+1)*8 +1 = 137 processes

MPI BLAST

- As noted by mpiblast developers
  - Configurations of partition-size, number-of-workers, etc. are platform/ workload dependent
  - Fine-tuning and testing are necessary to achieve optimal performance

Statistics of Distributed Database Searches

STATISTICS OF DISTRIBUTED DATABASE SEARCHES

- What issues arise from fragmenting a database?

STATISTICS OF DISTRIBUTED DATABASE SEARCHES

- As database size gets smaller, the chance of identifying sequences with same score is less (more significant)

  - If we fragment the database we are generating statistics on smaller databases

STATISTICS OF DISTRIBUTED DATABASE SEARCHES

- BLAST parameter “-dbsize”
- Effective length
  - The effective length of the database is the result of dividing the effective length of the search space by the effective length of the query
    - Default value is 0, which means the real length of the database

STATISTICS OF DISTRIBUTED DATABASE SEARCHES

- Search Space
  - Represents all sites at which query can align to database sequences
- Effective Search Space
  - Ends of a sequence are less likely to align in an average-sized alignment (edge effect)
  - Effective search space = (m-L)(n-L)
  - L = length of an average sized alignment (Altschul and Gish, 1996)
    - Calculate dynamically

MPIBlast vs Blast+

MPIBLAST VS BLAST+

- Which to use?
- NCBI BLAST allows multi-threading (having multiple workers), but within one node
  - Limitations of design
  - Limitations of implementation (only partially multi-threaded)
- mpiBLAST allows multi-threading across multiple across nodes
  - Adds other essential improvements such as dynamic load balancing and optimization of I/O operations
  - Overhead

MPIBLAST VS BLAST+

- Of course “it depends”
- Contentious debate on best practices
  - Google it
- My Opinion
  - Which ever is easiest to organize and debug
  - Having to run it twice will always be slower
  - Smart preprocessing is always required

MPIBLAST VS BLAST+

- NCBI Architecture
  - Many unrelated tasks
  - Tasks are multiprogrammed on collection of servers
- NCBI example
  - NCBI maintains cluster for GenBank
  - Web server receives request to “blast” sequences X, Y, Z…
    - Farms out individual requests to separate PCs
    - Collects answer and create web page with result
- As size of sequence databases grow they may need to exploit parallelism for individual applications


PROTEIN
FUNCTION

PROTEIN MACHINES

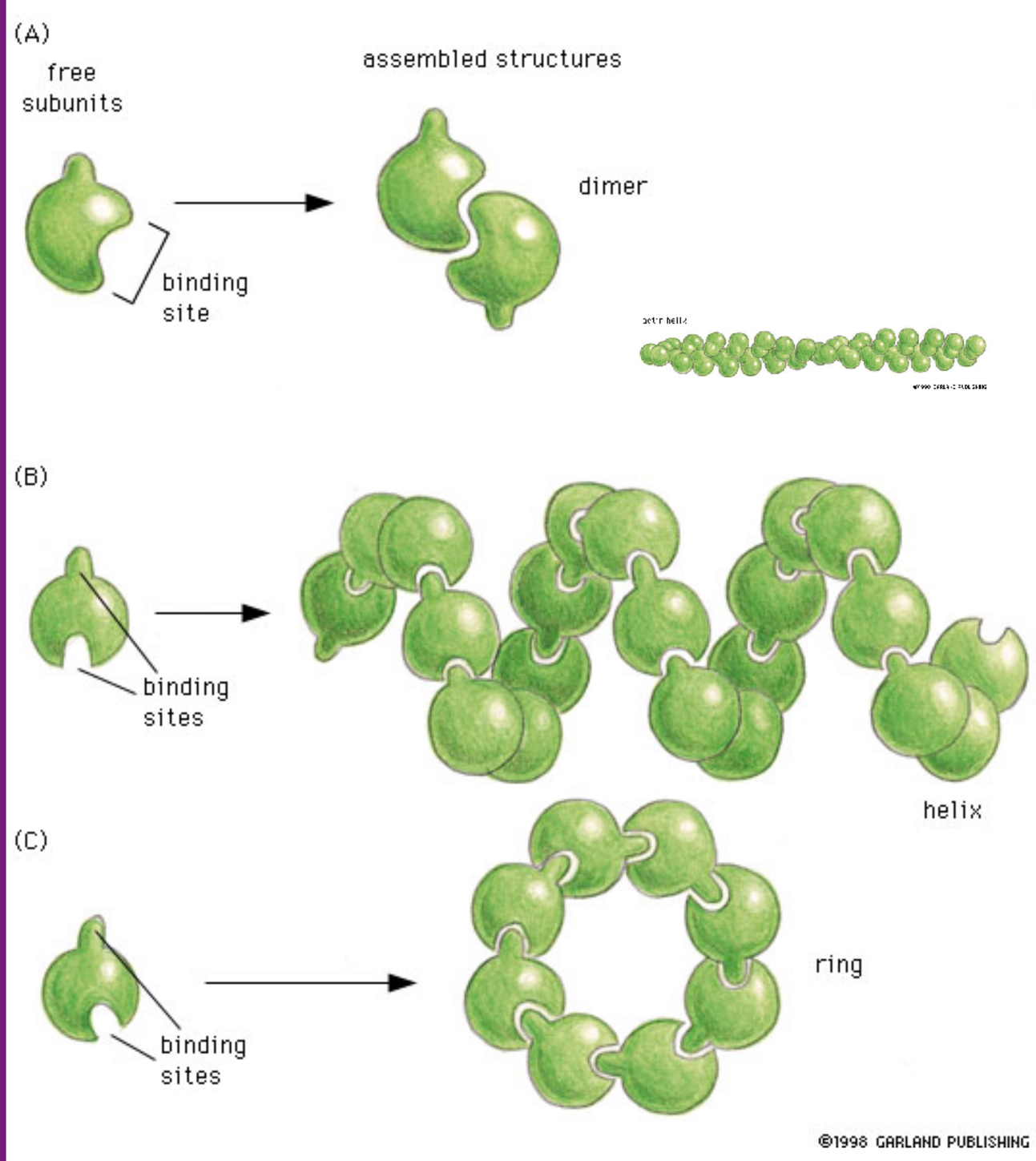
- Molecular Machines of Life (http://www.youtube.com/watch?v=FJ4N0iSeR8U)

PROTEIN FUNCTION

- Proteins
  - Make up about 15% of the cell
- Workhorses
  - Enzymes
  - Structural
  - Transport
  - Motor
  - Storage
  - Signaling
  - Receptors
  - Gene regulation
  - Special functions

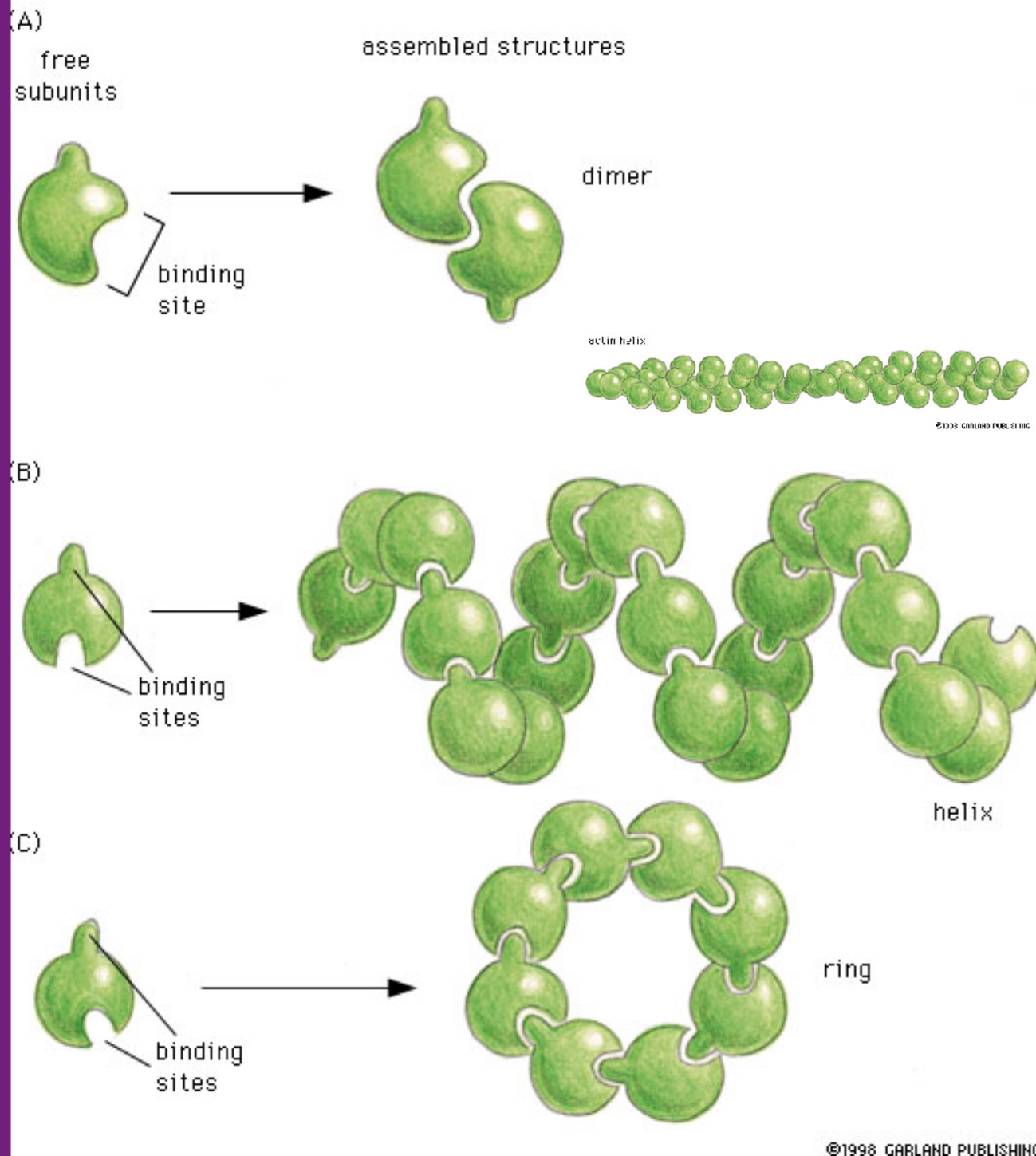
PROTEIN FUNCTION

- Globular Proteins
  - Most of what we have dealt with so far
  - Compact shape like a ball with irregular surfaces
    - Swiss cheese
  - Enzymes are globular



PROTEIN FUNCTION

- Globular protein functions
  - Storage of ions and molecules
    - myoglobin, ferritin
  - Transport of ions and molecules
    - hemoglobin, serotonin transporter
  - Defense against pathogens
    - antibodies, cytokines
  - Muscle contraction
    - actin, myosin
  - Biological catalysis
    - chymotrypsin, lysozyme

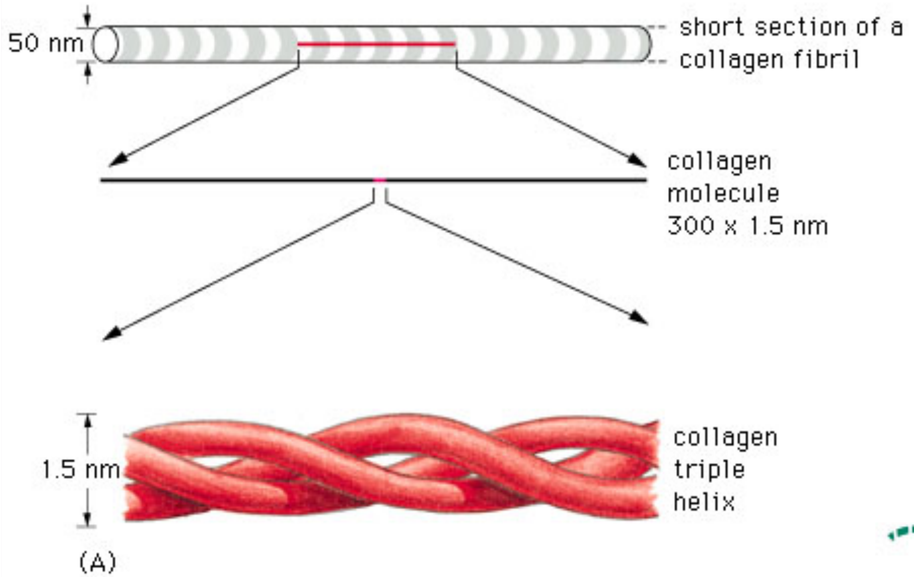


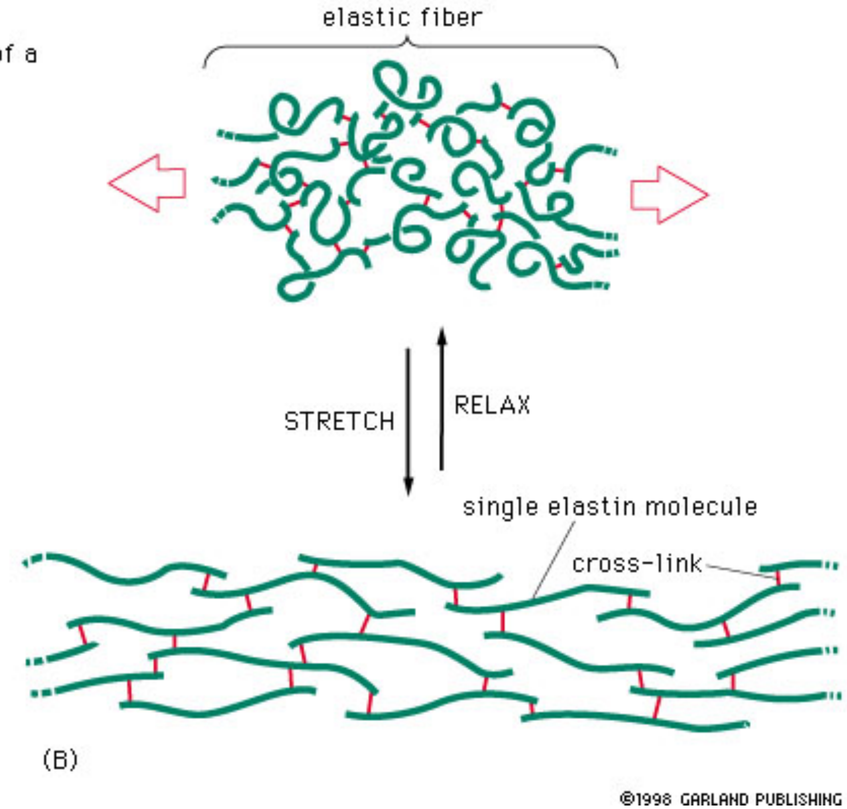
PROTEIN FUNCTION

- Fibrous Proteins
  - Span a long distance in the cell
  - 3-D structure is usually long and rod shaped
- Important fibrous proteins
  - Intermediate filaments of the cytoskeleton
    - Structural scaffold inside the cell
      - Keratin in hair, horns and nails
  - Extracellular matrix
    - Bind cells together to make tissues
    - Secreted from cells and assemble in long fibers

PROTEIN FUNCTION

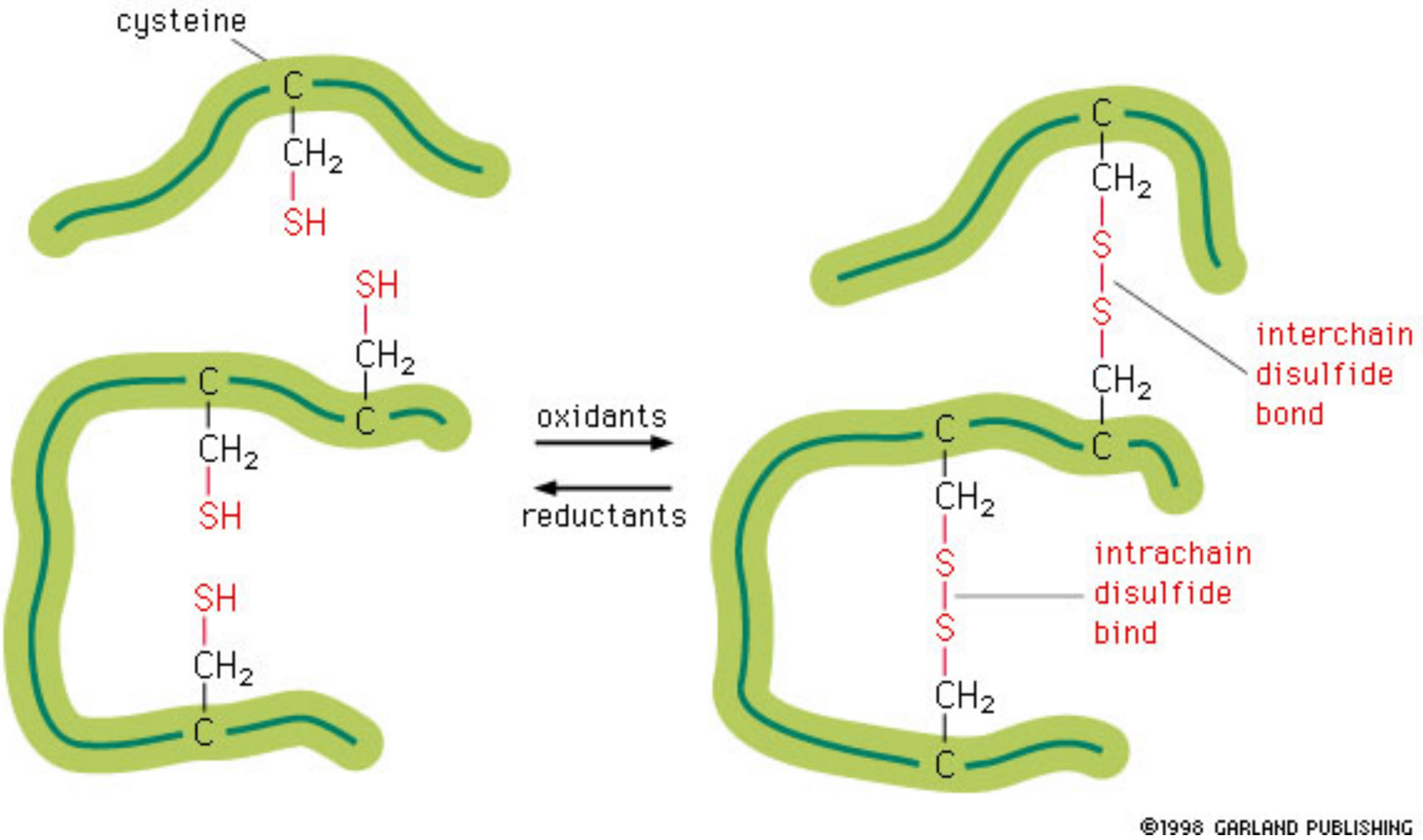
- Collagen
  - Fiber with a glycine every third amino acid in the protein
- Elastin
  - Unstructured fibers that gives tissue an elastic characteristic





PROTEIN FUNCTION

- Stabilizing Cross-Links
  - Cross linkages can be between 2 parts of a protein or between 2 subunits
  - Disulfide bonds (S-S) form between adjacent -SH groups on the amino acid cysteine



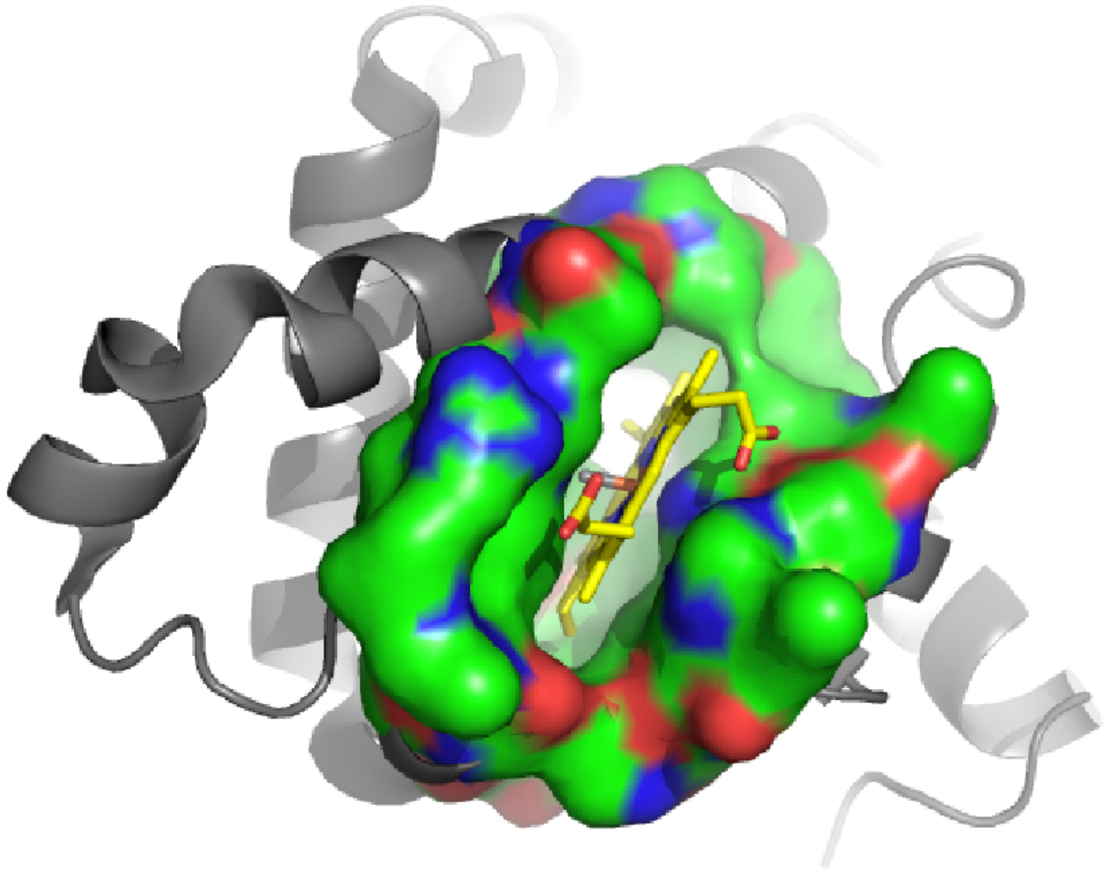
PROTEIN FUNCTION

- Walking along nanotubule
  - http://www.youtube.com/watch?v=-7AQVbrmzFw&list=PL51br4wRiaYYENlCVFY6q3Xz4FOMd97k9

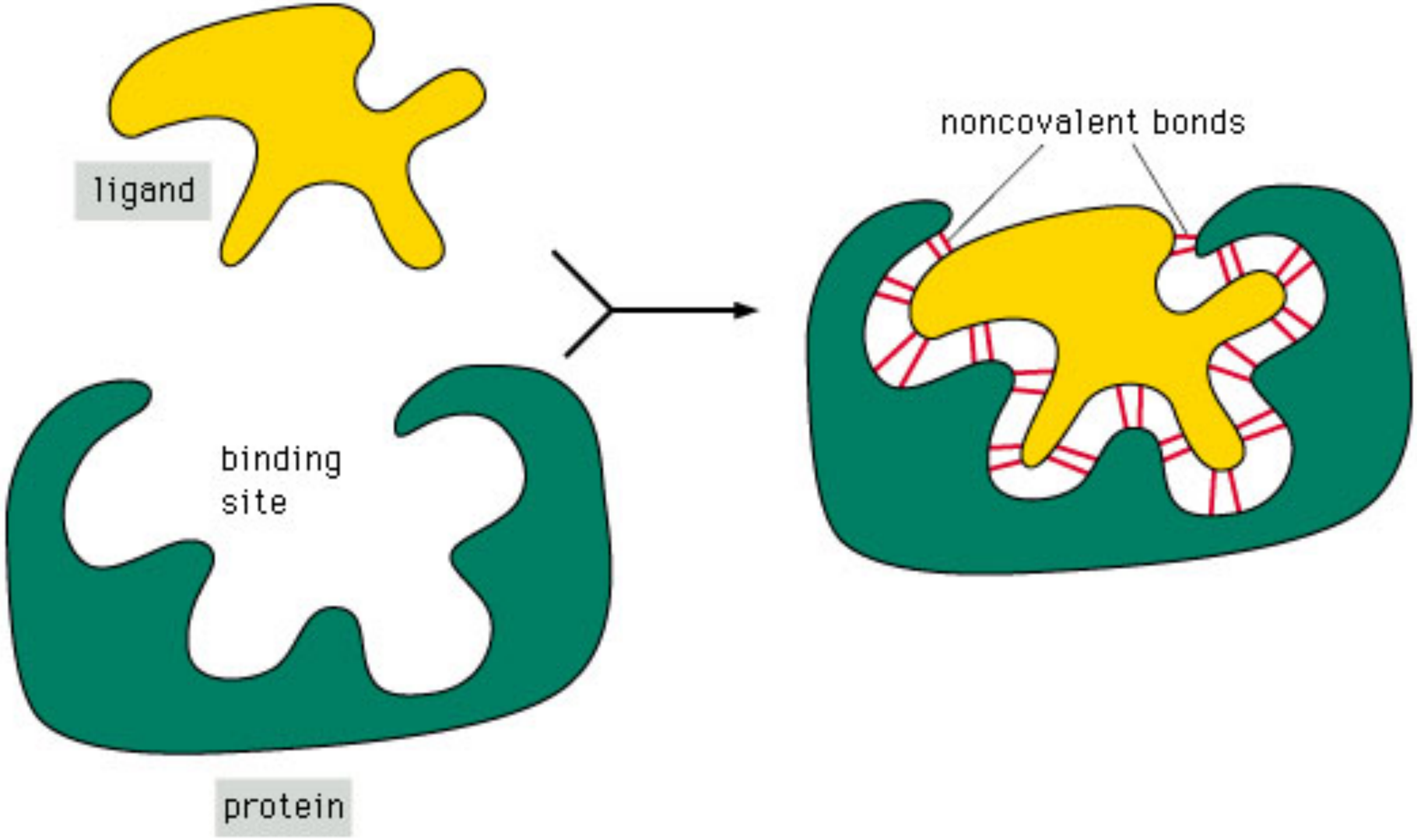
Ligand Binding

LIGAND BINDING

- The conformation of a protein gives it a unique function
  - Proteins interact with other molecules
- Ligand
  - Molecule that a protein can bind
- Binding site
  - Part of the protein that interacts with the ligand
  - Consists of a cavity formed by a specific arrangement of amino acids

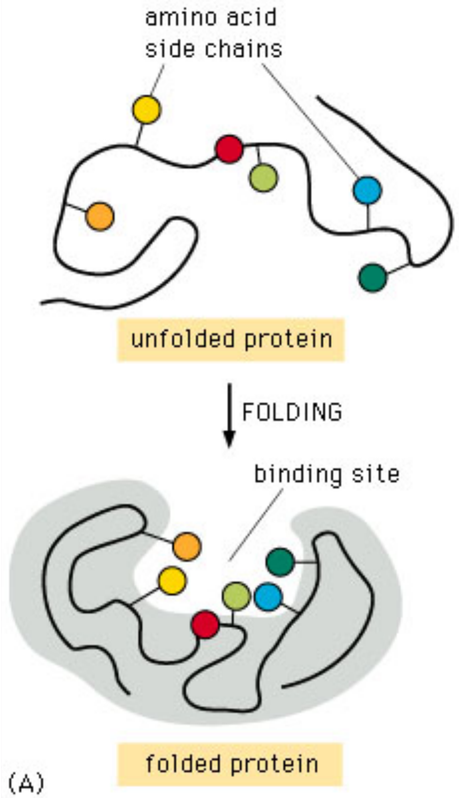


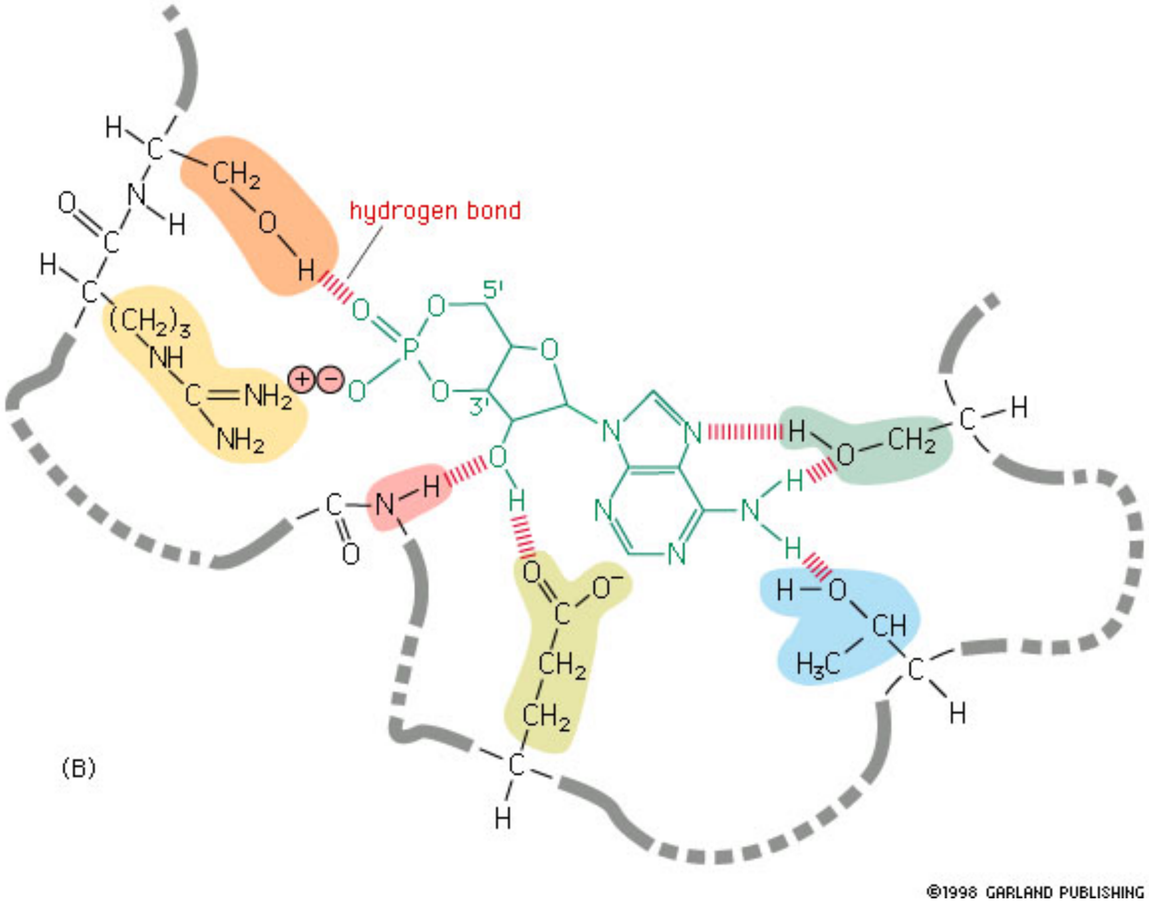
LIGAND BINDING



LIGAND BINDING

- Formation of Binding Site
  - Amino acids that come together in the folding
  - The remaining sequences may play a role in regulating the protein's activity





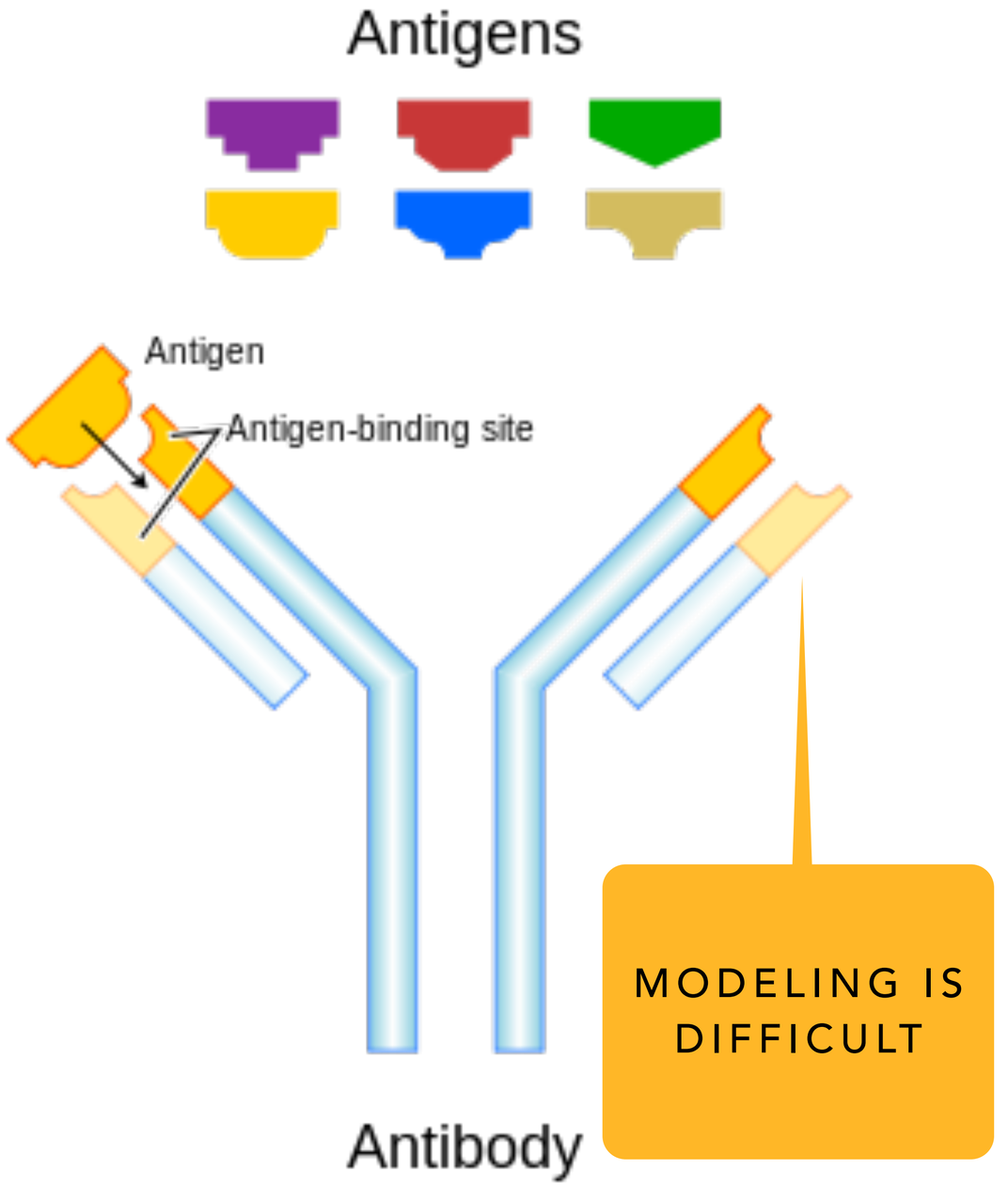
LIGAND BINDING

- Antibody
  - A family of proteins that can be created to bind to almost any molecule
  - Antibodies (immunoglobulins) are made in response to a foreign molecule
    - Bacteria, virus, pollen
    - Called the antigen
  - Bind together tightly and inactivates the antigen
    - Marks it for destruction

RESPONSE TO INVADERS

LIGAND BINDING

- Antibody
  - Y-shaped molecules with 2 binding sites at the upper ends of the Y
  - The loops of polypeptides on the end of the binding site are what imparts the recognition of the antigen
  - Changes in the sequence of the loops make the antibody recognize different antigens
    - Specificity

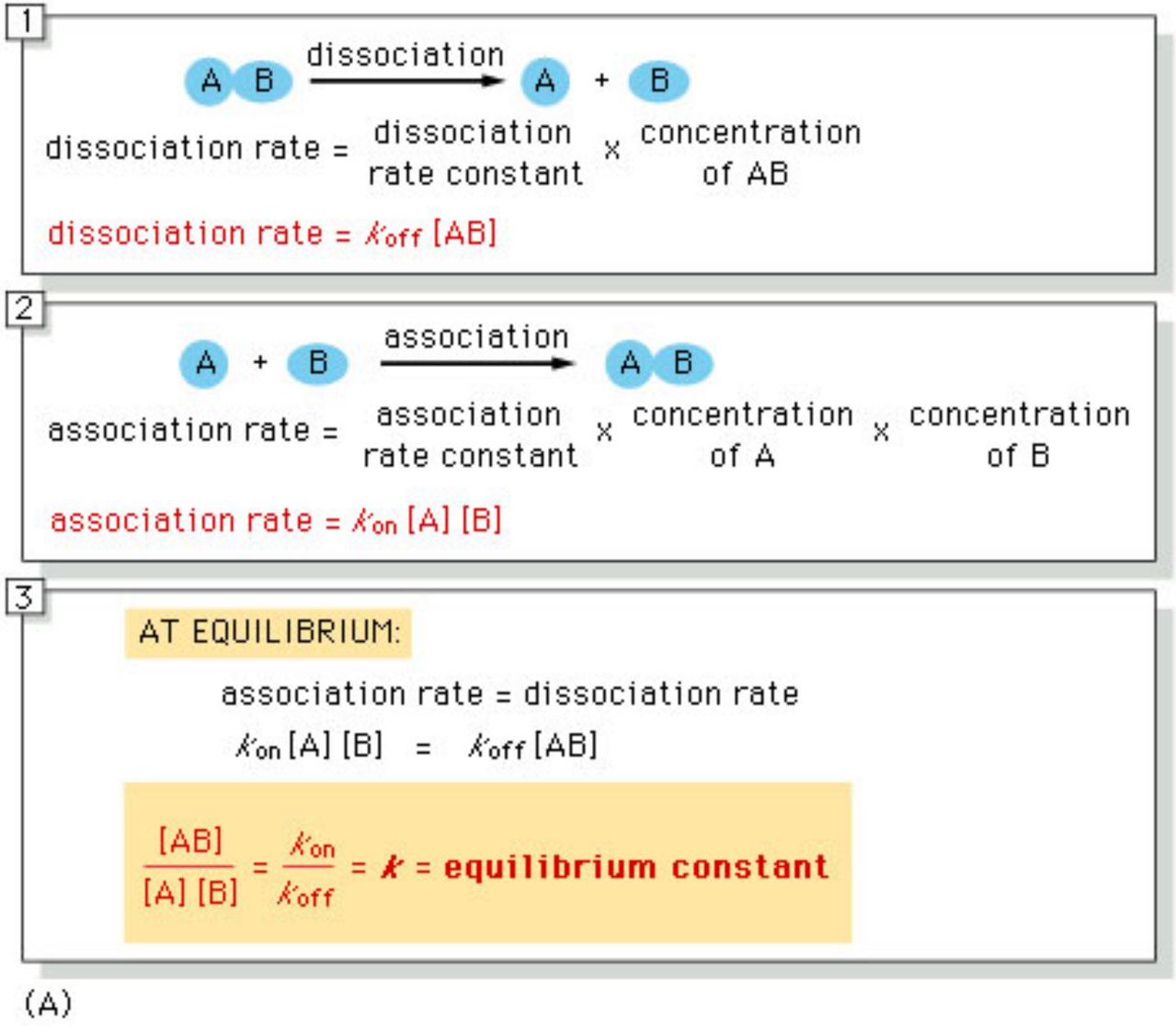


LIGAND BINDING

- Binding strength
  - Can be measured directly
  - Antibodies and antigens are mixing around in a solution, eventually they will bump into each other in a way that the antigen sticks to the antibody, eventually they will separate due to the motion in the molecules
  - This process continues until the equilibrium is reached – number sticking is constant and number leaving is constant
  - This can be determined for any protein and its ligand

LIGAND BINDING

- Equilibrium constant
  - Concentration of antigen, antibody and antigen/antibody complex at equilibrium can be measured – equilibrium constant (K)
  - Larger the K the tighter the binding or the more non-covalent bonds



(A)

Enzymes

ENZYMES

- Enzymes
  - Proteins that bind to their ligand as the first step in a process
  - Substrate
    - An enzyme’s ligand
    - May be 1 or more molecules
  - Product
    - Output of the reaction
  - Enzymes can repeat these steps many times and rapidly, called catalysts

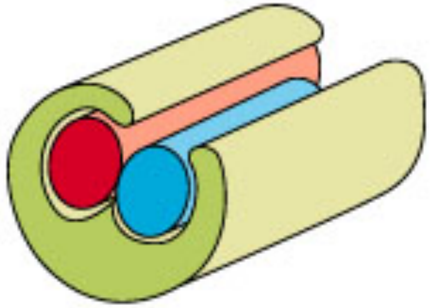
ENZYMES

- Active site
  - Special binding site in enzymes where the chemical reaction takes place
  - Catalytic triad
- Transition state
  - State between substrate and product

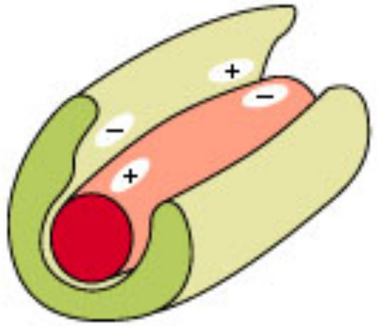
- Bonds
  - Covalent
    - Share electron pairs; highly stable
  - Non-covalent
    - Allows the interactions to be transient
    - Ionic bonds
    - Hydrogen bonds
      - Electrostatic interactions between polar atoms (and bound H)
    - Hydrophobic

ENZYMES

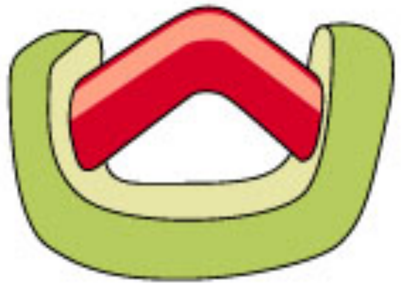
- Different mechanisms of enzyme catalysis



(A) enzyme binds to two substrate molecules and orients them precisely to encourage a reaction to occur between them



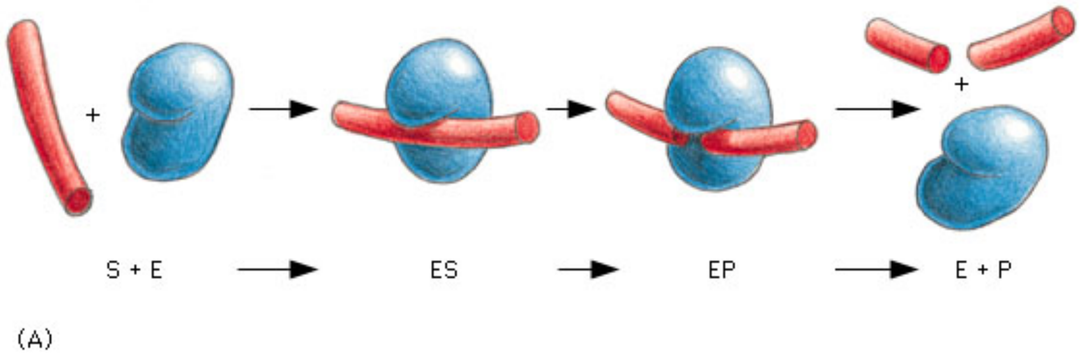
(B) binding of substrate to enzyme rearranges electrons in the substrate, creating partial negative and positive charges that favor a reaction

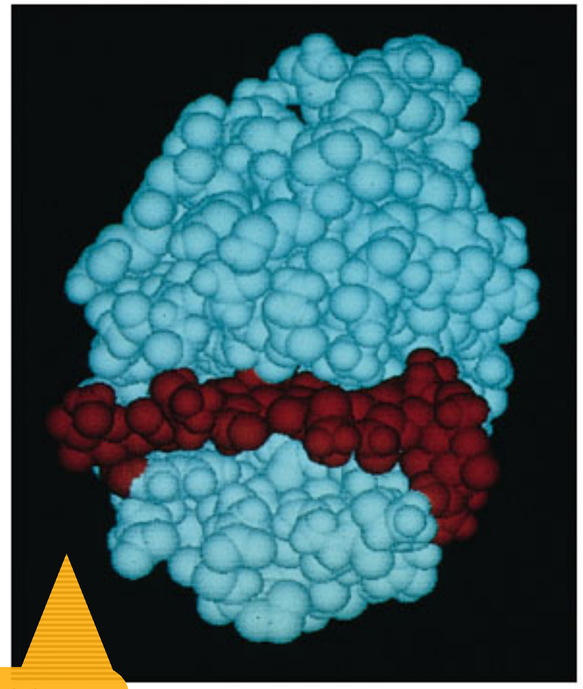


(C) enzyme strains the bound substrate molecule, forcing it toward a transition state to favor a reaction

ENZYMES

- Lysozyme
  - Protects us from bacteria by making holes in the bacterial cell wall and causing it to break
  - Adds H2O to the glycosidic bond in the cell wall
  - Easy protein to crystallize (used for benchmarking detectors)





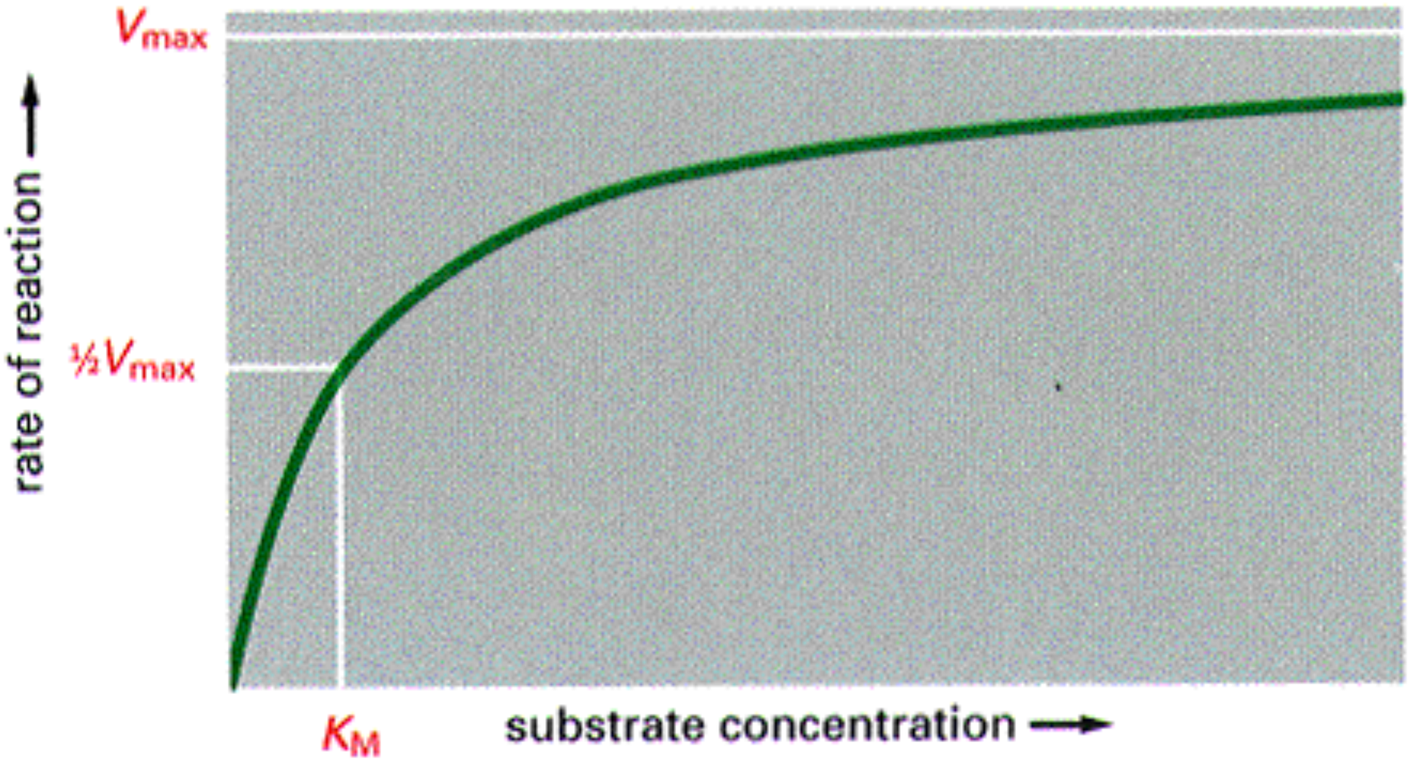
NON-COVALENT BONDS HOLD THE POLYSACCHARIDE IN THE ACTIVE SITE UNTIL THE REACTION OCCURS

ENZYMES

- Enzyme Performance
  - $E + S \leftrightarrow ES \leftrightarrow EP \leftrightarrow E + P$
- Step 1: Binding of the substrate
  - Limiting step depending on [S] and/or [E]
  - Vmax – maximum rate of the reaction
  - Turnover number determines how fast the substrate can be processed = rate of rxn ÷ [E]
- Step 2: Stabilize the transition state
  - State of substrate prior to becoming product
  - Enzymes lowers the energy of transition state and therefore accelerates the reaction

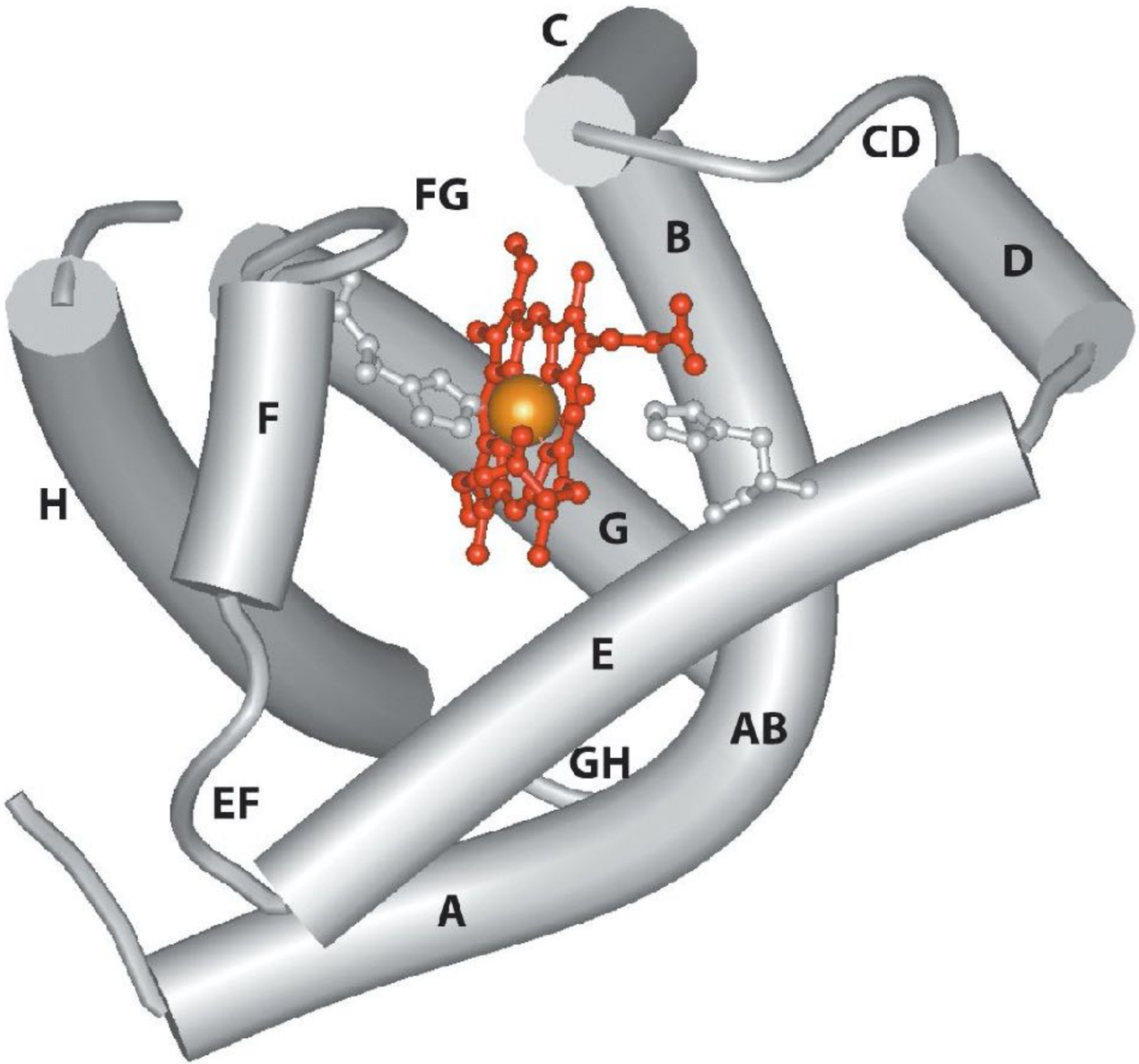
ENZYMES

- K_M
  - [S] that allows rate of reaction to proceed at ½ it maximum rate



PROSTHETIC GROUPS

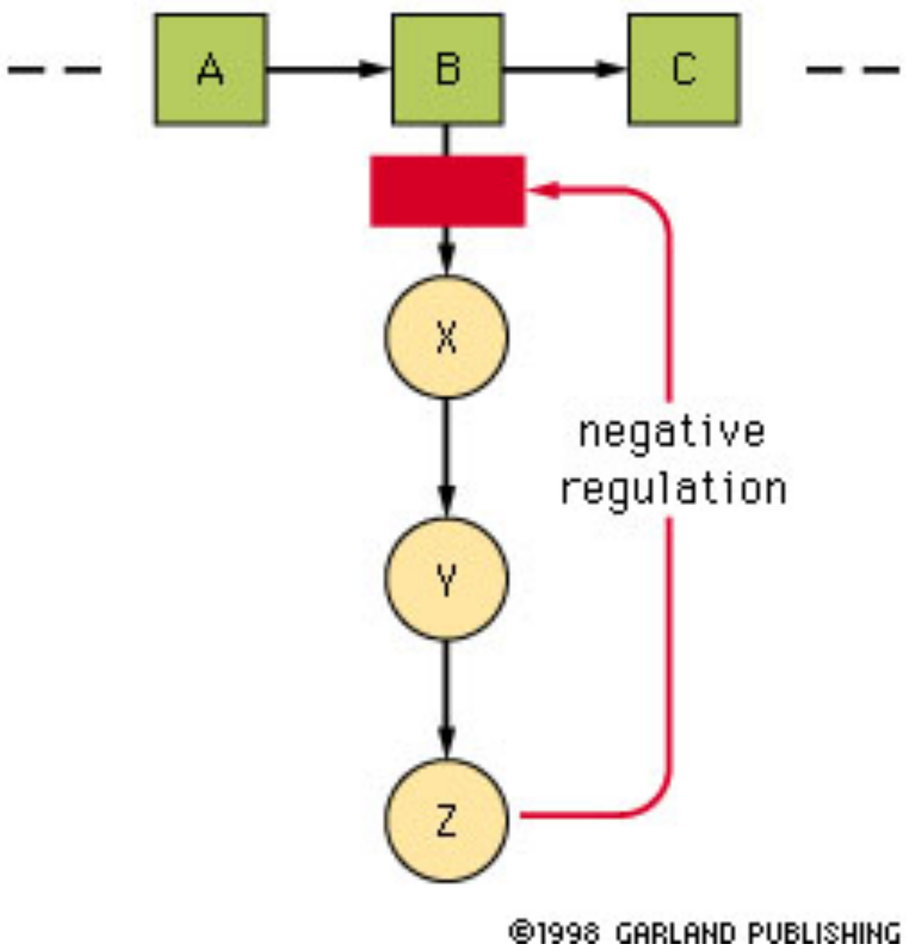
- Some proteins require a non-protein molecule to enhance the performance of the protein
  - Hemoglobin requires heme (iron containing compound) to carry the O2
- When a prosthetic group is required by an enzyme it is called a co-enzyme
  - Typically a metal or vitamin
  - These groups may be covalently or non-covalently linked to the protein



Enzymatic Regulation

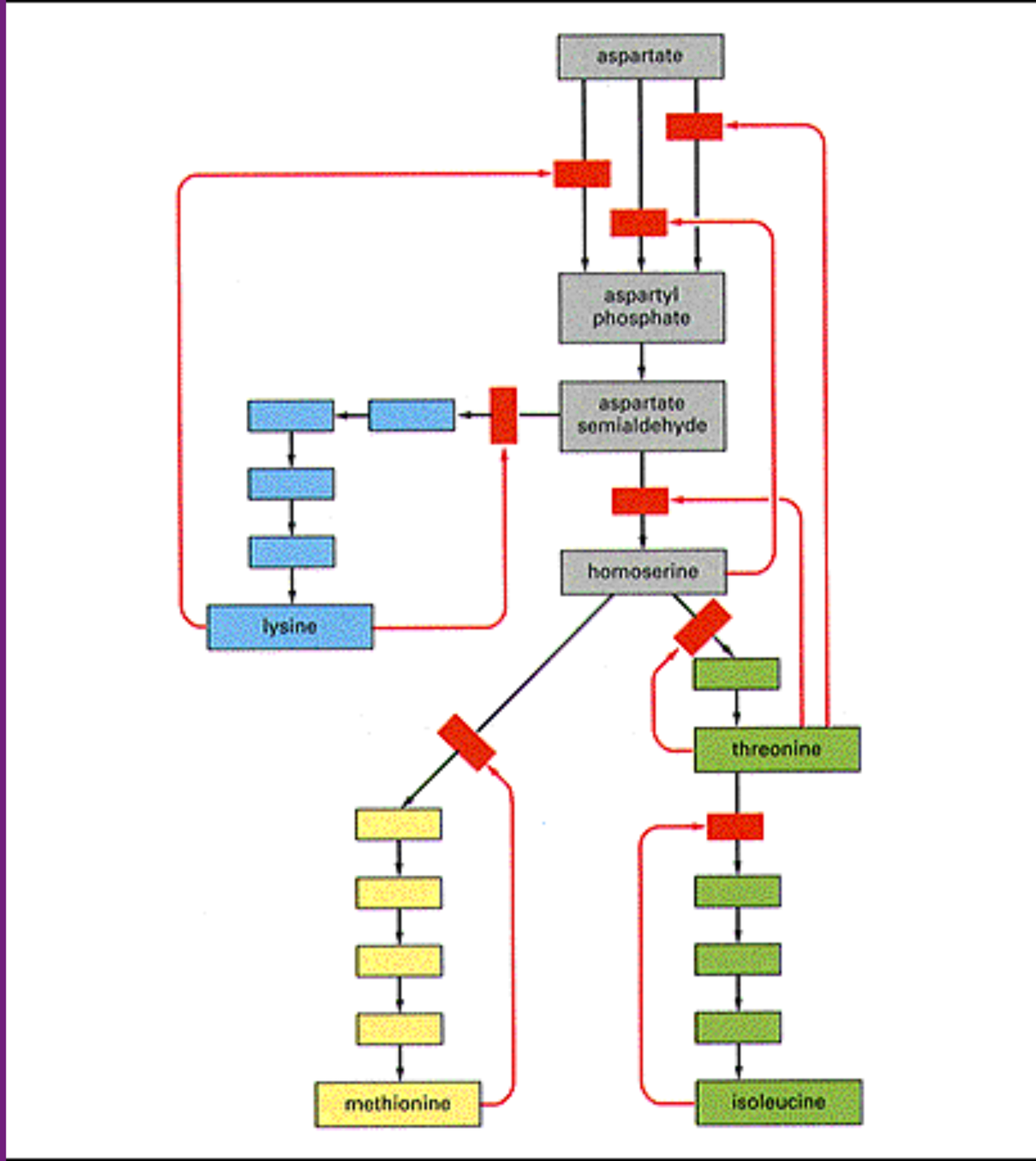
ENZYMATIC REGULATION

- Regulation of enzymatic pathways prevent the deletion of substrate
- Regulation happens at the level of the enzyme in a pathway
- Feedback inhibition is when the end product regulates the enzyme early in the pathway



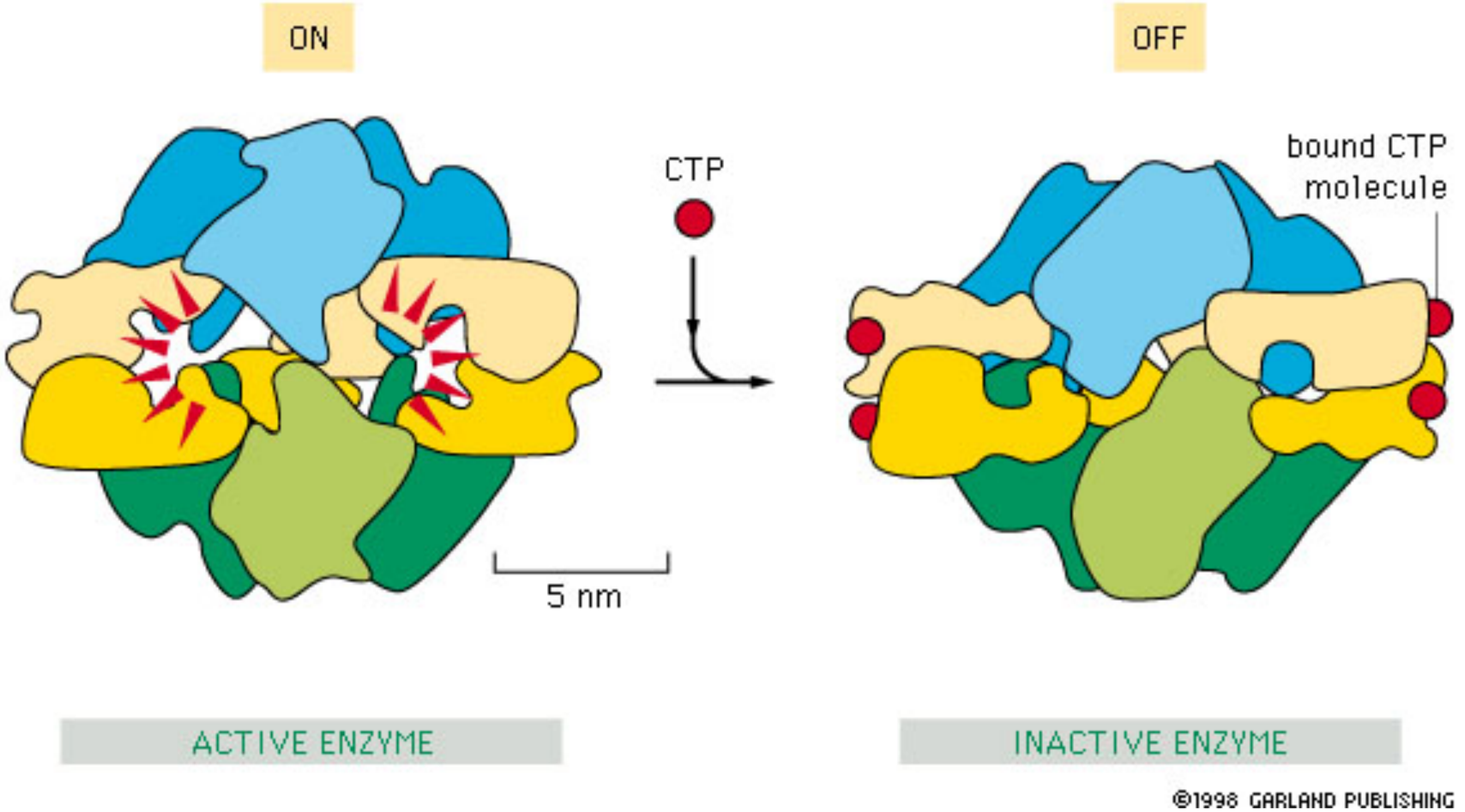
ENZYMATIC REGULATION

- Negative feedback
  - Pathway is inhibited by accumulation of final product
- Positive feedback
  - A regulatory molecule stimulates the activity of the enzyme
    - Typically between two pathways
    - ↑ ADP levels cause the activation of the glycolysis pathway to make more ATP



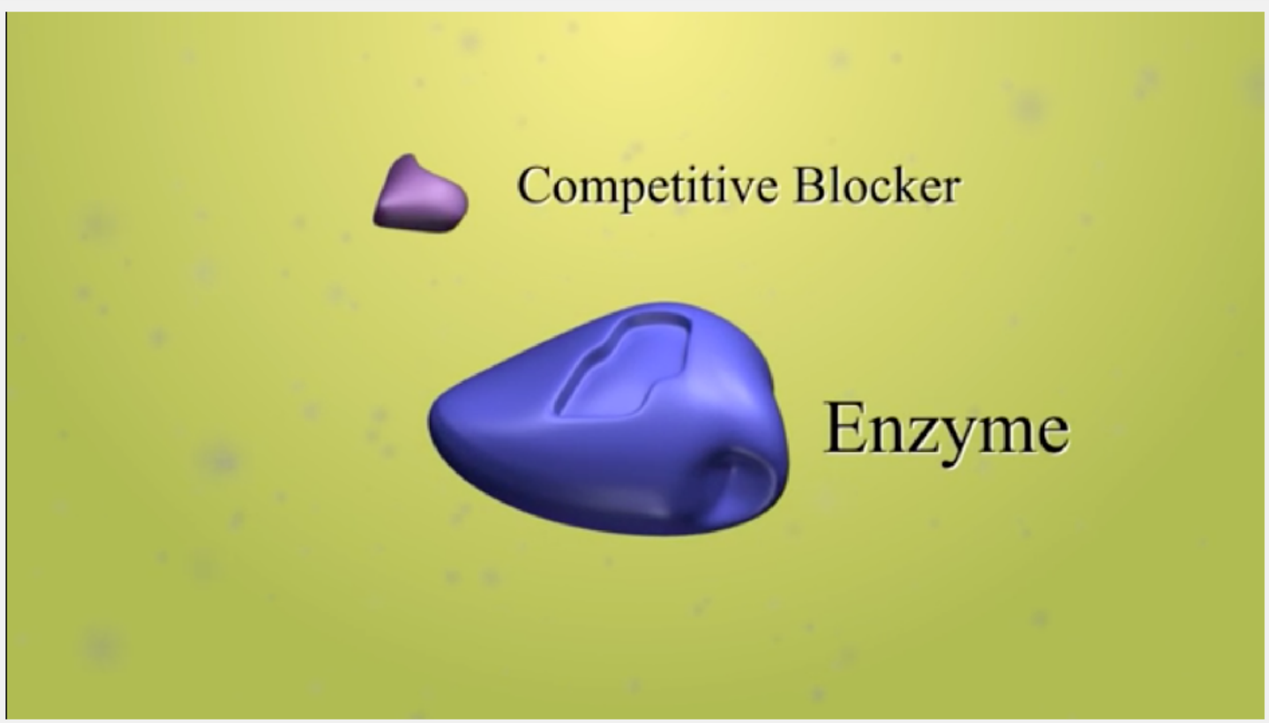
ENZYMATIC REGULATION

- Allosteric regulation
  - Method of regulation is also used in other proteins besides enzymes
  - Receptors, structural and motor proteins



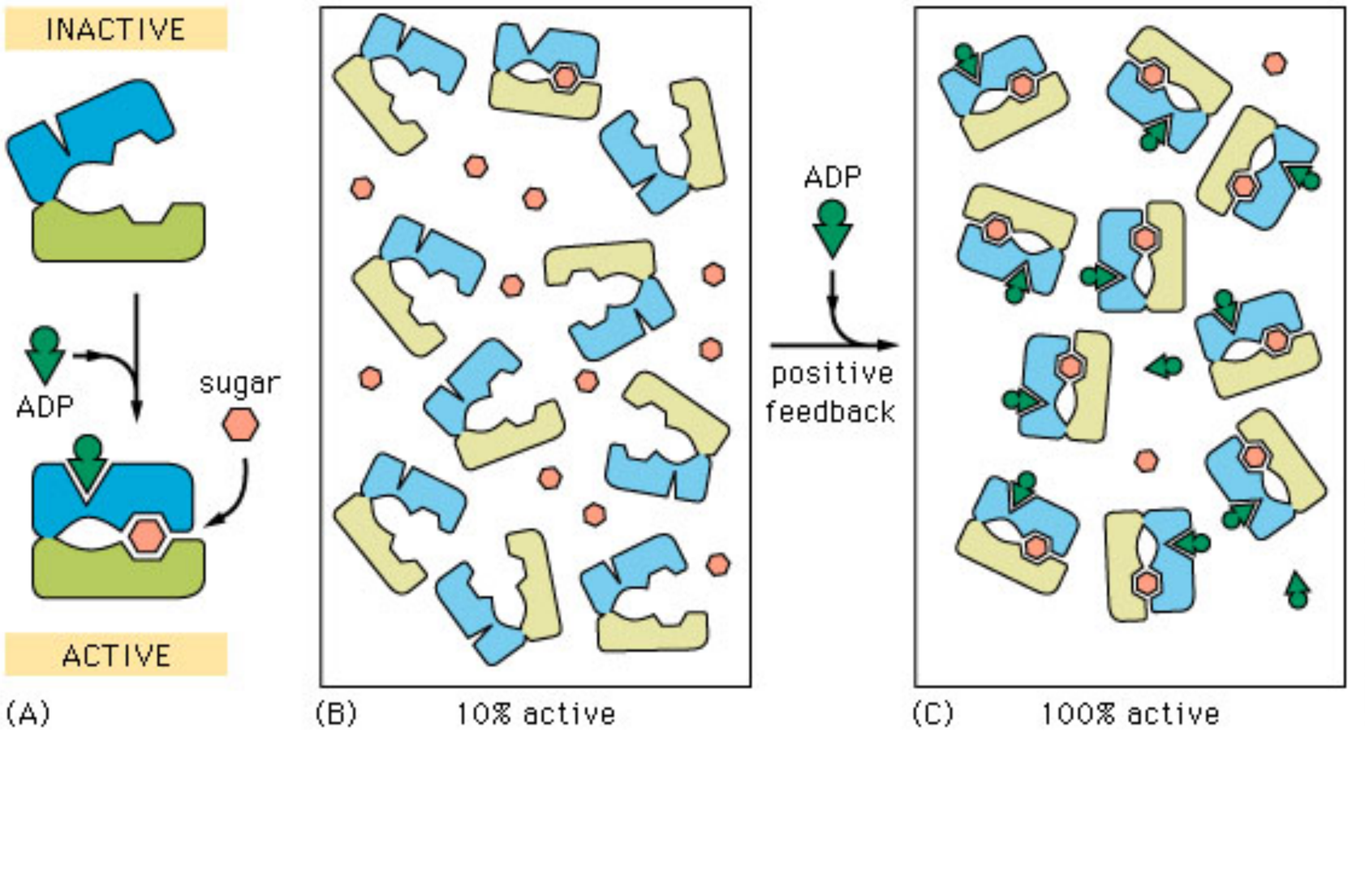
ENZYMATIC REGULATION

- Allosteric regulation
  - Conformational coupling of widely separated binding sites
  - Active site recognizes substrate and 2nd site recognizes the regulatory molecule
  - Protein regulated this way undergoes allosteric transition or a conformational change
- Video:
  - http://www.youtube.com/watch?v=PILzvT3spCQ



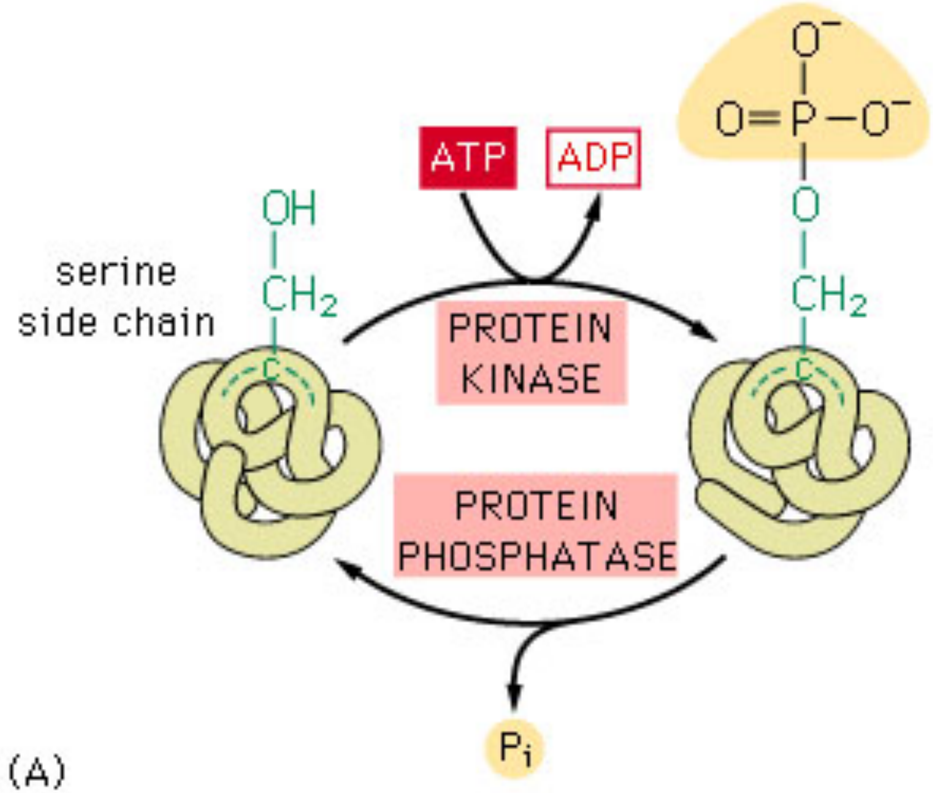
ENZYMATIC REGULATION

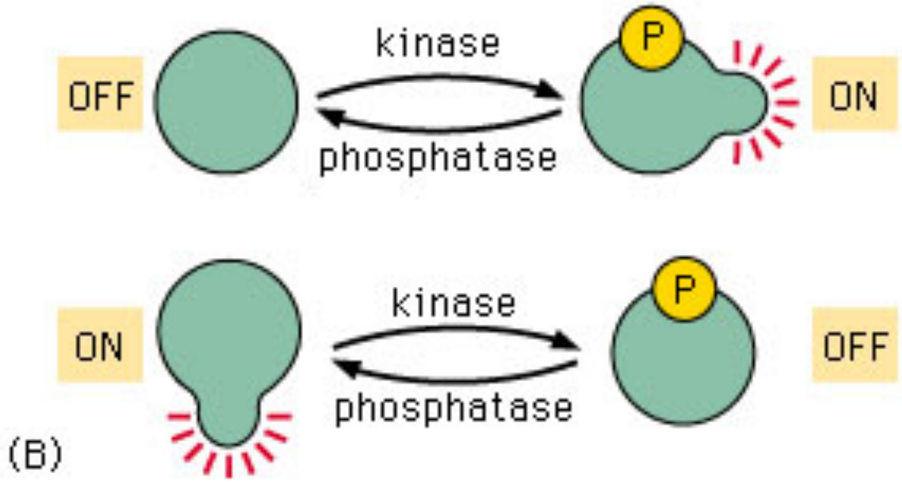
- Example
  - Enzyme is only partially active with sugar only but much more active with sugar and ADP present



ENZYMATIC REGULATION

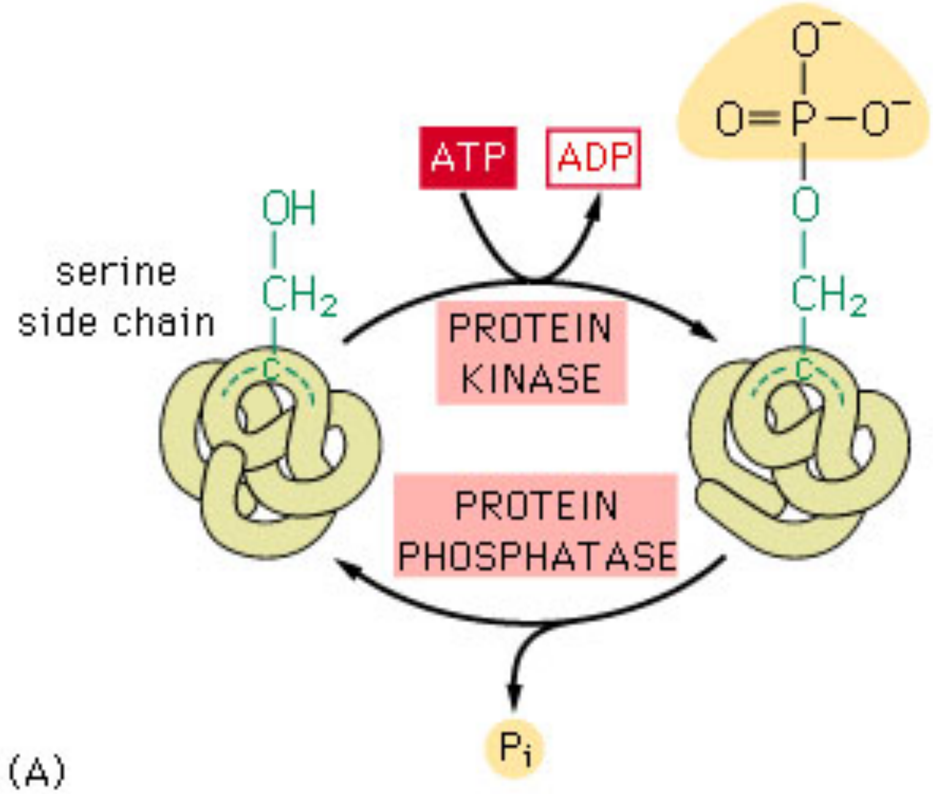
- Phosphorylation
  - Regulation by the addition of a phosphate group that allows for the attraction of + charged side chains causing a conformation change
  - Regulate many eukaryotic cell functions turning things on and off
  - Protein kinases add the phosphate
  - Protein phosphatase remove them

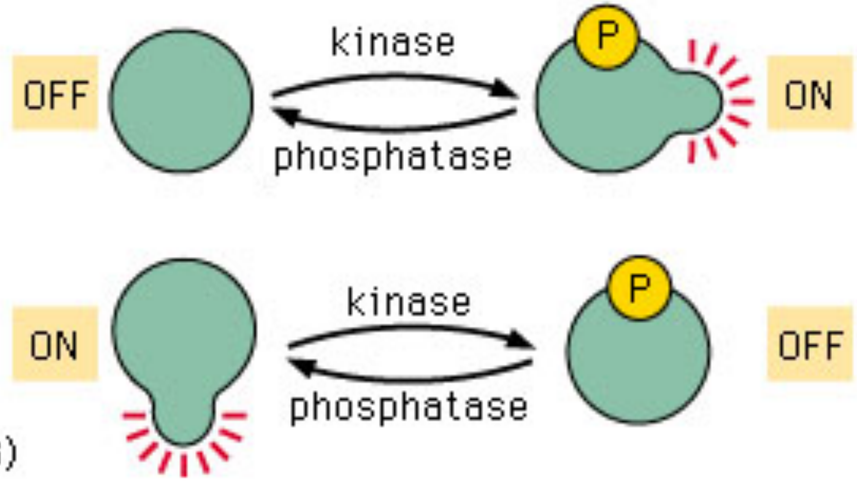




ENZYMATIC REGULATION

- Phosphorylation
  - Kinases capable of putting the PO4 on 3 different amino acid residues
  - Have a –OH group on R group
    - Serine
    - Threonine
    - Tyrosine
- Phosphatases that remove the PO4 may be specific for 1 or 2 reactions or many be non-specific



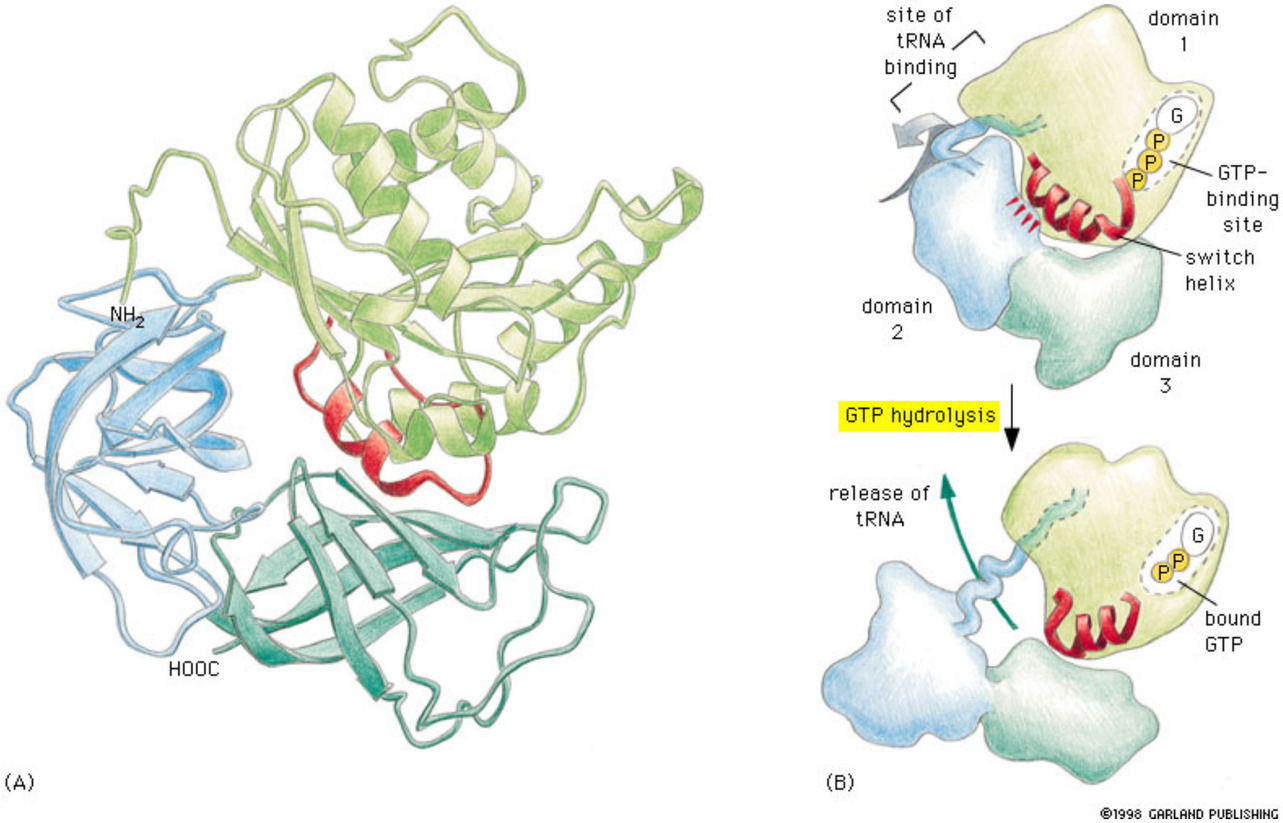


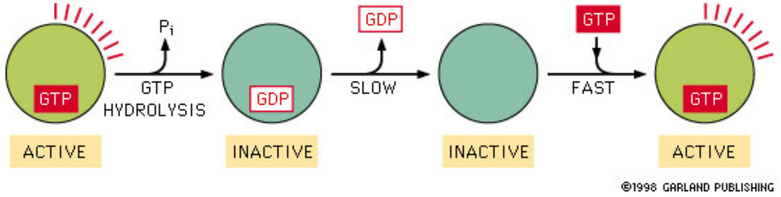
ENZYME REGULATION

- MAPK Signalling Pathway
  - http://www.youtube.com/watch?v=oDjDUUhGVsI

ENZYMATIC REGULATION

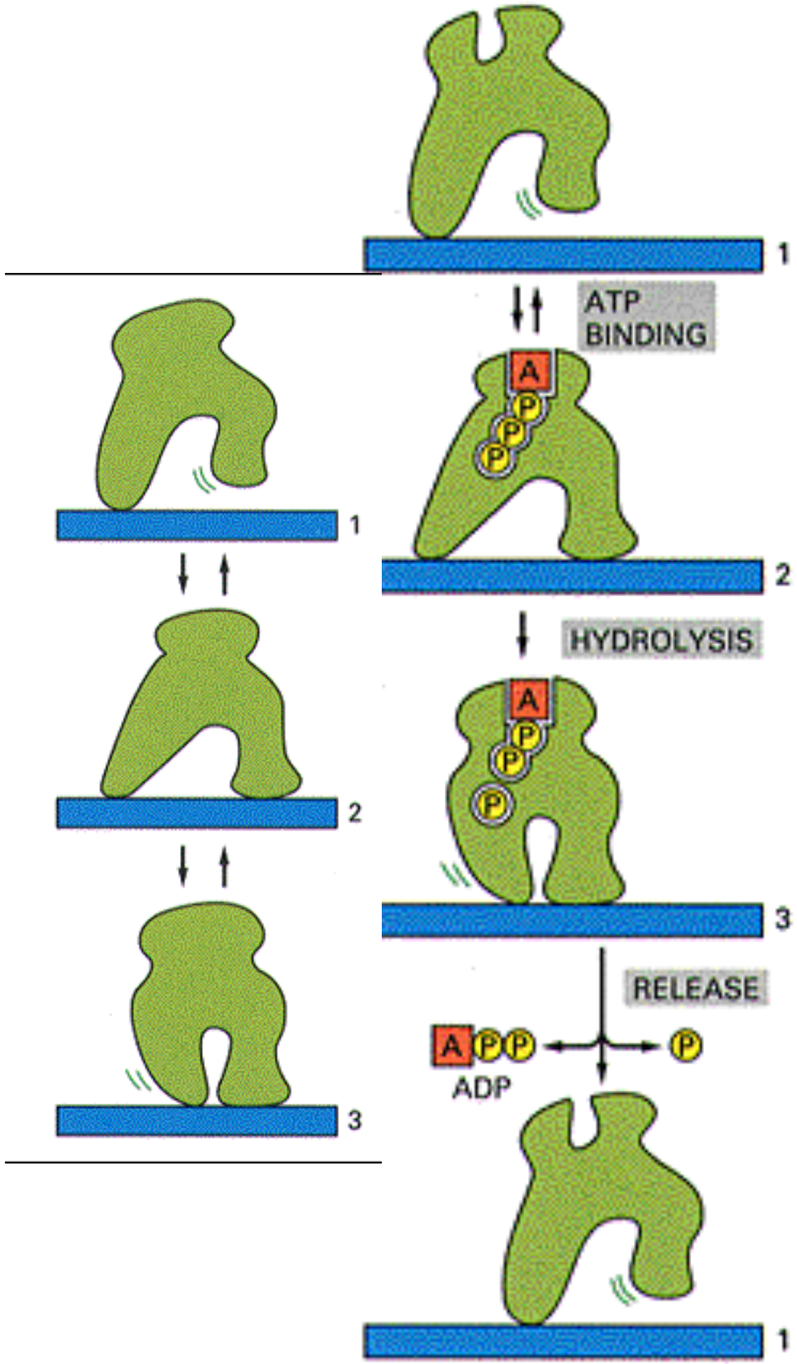
- Molecular switches
- GTP-Binding Proteins (GTPases)
  - GTP does not release its PO4 group but rather the guanine part binds tightly to the protein and the protein is active
  - Hydrolysis of the GTP to GDP (by the protein itself) and now the protein is inactive
  - Also a family of proteins usually involved in cell signaling switching proteins on and off





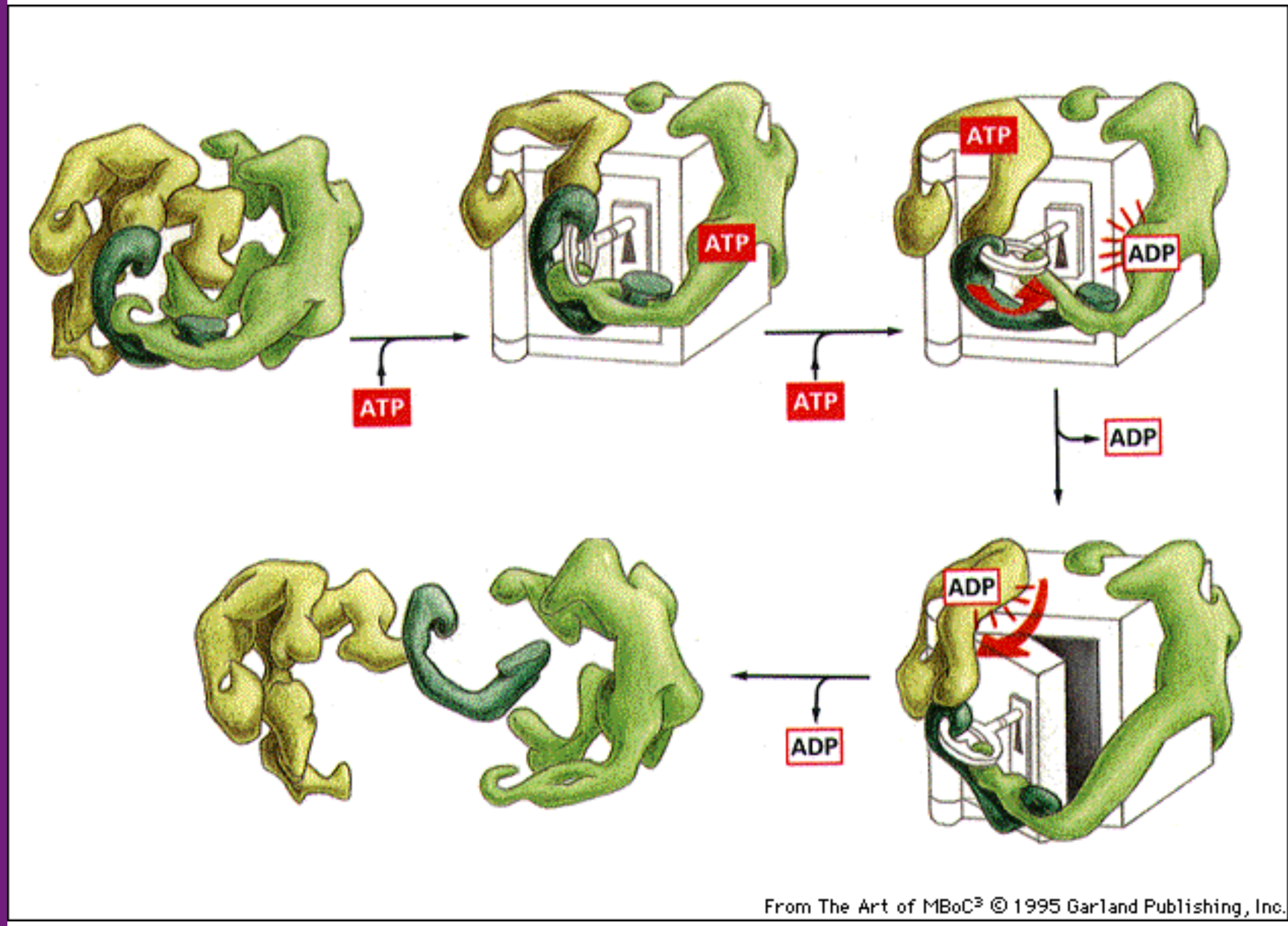
ENZYMATIC REGULATION

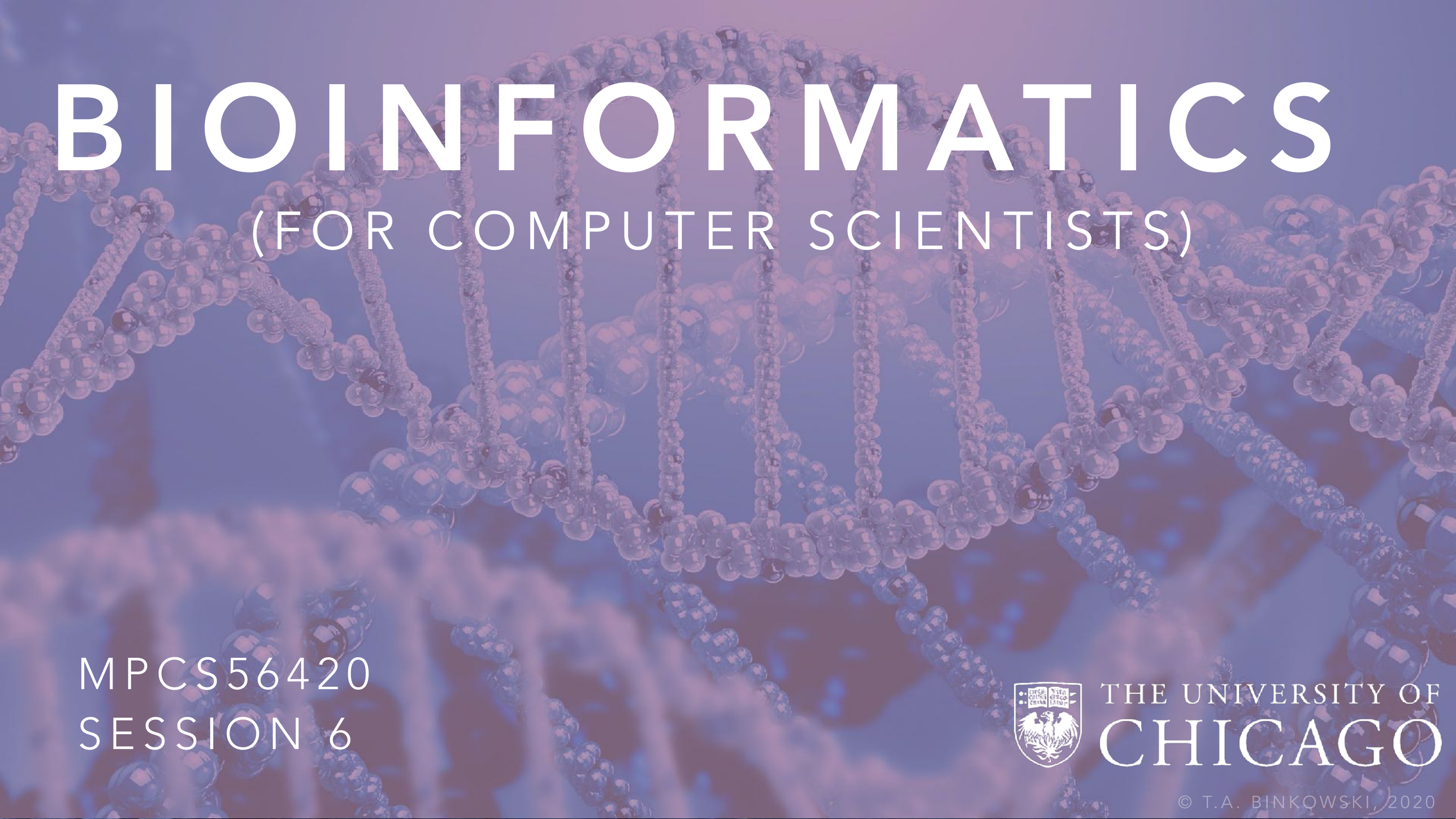
- Motor Proteins
  - Proteins can move in the cell, say up and down a DNA strand but with very little uniformity
  - Adding ligands to change the conformation is not enough to regulate this process
  - The hydrolysis of ATP can direct the the movement as well as make it unidirectional
  - The motor proteins that move things along the actin filaments or myosin



PROTEIN MACHINES

- Complexes of 10 or more proteins that work together such as DNA replication, RNA or protein synthesis, trans-membrane signaling etc.
    - Usually driven by ATP or GTP hydrolysis





BIOINFORMATICS

(FOR COMPUTER SCIENTISTS)

MPCS56420
SESSION 6

THE UNIVERSITY OF CHICAGO