

# Diagrammatic Notation and Computational Structure of Gene Networks

Ron Maimon and Sam Browning  
Gene Network Sciences  
2359 N. Triphammer Rd.  
Ithaca, New York

ron@gnsbiotech.com sam@gnsbiotech.com

## ABSTRACT

A formal notation for gene networks is introduced. This notation makes it possible to describe all the interactions in a cell in a single diagram, with only a few representations of each molecule. The notation is compact, introducing several ideas borrowed from computational mathematics into biology. It is modular, in the sense that complex interactions composed of many subparts may be annotated with the same symbols as the simplest interactions composed of individual molecules or genes.

## 1. BIOLOGY IS COMPUTATION

The behavior of biological systems may be best understood as an abstract computation. The units of biological heredity are packets of information, and the cell's metabolic machinery is a layer of computation evolved with the goal of replicating the data stored in the hereditary material.

Although the basic operations of a cell are the joining and splitting of molecules, there is a strong analogy between this process and the process of flipping and copying bits of information in a computer's memory. Both processes are simple in isolation, but when integrated into a network can become complex enough to produce arbitrarily complex behavior.

The basic interactions of metabolism differ from the basic interactions of an electronic computer because the cell has many different types of molecules, and each type carries information about the state of the cell. The electronic computer has a homogeneous information carrier, the bit, and the fundamental interaction is uniform, a logical NAND gate.

So when modelers aim to describe biological networks, the language that we use must reflect the peculiar nature of the computational machinery of the cell.

The basic interactions of molecules, the reactions that are analogous to the logical operations, are the binding and unbinding of molecule to molecule. Previous attempts to provide a standard notation [1] for interaction networks represented interactions at the most basic level. In a computer, the analogous language would be composed of logical single-bit instructions only. This is sufficient for annotating simple networks, but very quickly it is apparent that a higher level of description is required for compactness and readability.

The notation described in this paper is an attempt to standardize a higher-level language for genetic networks, in the spirit of FORTRAN, LISP and C.

## 2. ATOMS AND ACTIONS

The smallest computational units in the description are *atoms*. These are not the chemists atoms, but are more like the ATOMs of LISP. They are indivisible units of biological function. A gene is an atom, as is a binding site on a complex protein. The entire protein may be an atom— if it can only bind one chemical at a time. An mRNA molecule is almost always an atom, since transcription and translation treat it as a unit, and these are the only actions the mRNA is known to participate in.

Not only molecules, but abstract conditions, such as metaphase, may be considered atoms. The two states Metaphase-on and Metaphase-off— although they indicate a complicated proportion and position of many molecules— are atoms. A process that causes metaphase to occur may be represented as the throwing of a switch, the transformation of one atom to another.

Some atoms are single-copy, an allele, or a condition switch, while others represent molecular species. Different types are modeled differently, so it is essential to distinguish the different types of atoms from each other.

We identify three basic types:

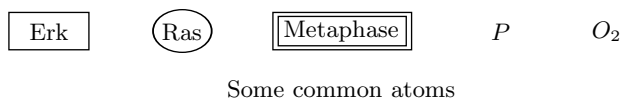
*Uniques*: these are atoms that are global and have a single occurrence. The centrosome, a Metaphase switch, or a single copy of a gene is a unique.

*Commons*: These atoms are found in many copies, but have dynamics. Proteins, mRNA, slow-diffusing molecules and calcium ions are all common atoms in many cells.

*Ubiques*: These are small atoms, which diffuse so quickly and come to equilibrium so quickly that they have no dynamics at all, but have a constant concentration in the cell. Phosphate groups, ATP, individual amino acids and nucleic acids, most ions, and small molecules in general, are ubiques.

A unique is represented by a double-line box, perhaps containing a name. Common atoms are ordinary boxes or cir-

cles. Ubiquies are represented by a chemical name without a box, or by a dashed circle or dashed box.

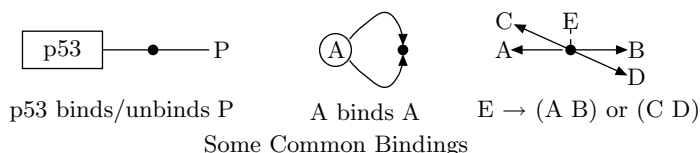


The atoms are the simplest nouns. The simplest verbs in the notation are the *actions*. Gene-network actions differ from the chemists' reactions, since they may be arbitrarily complex. This is true even though every action has sequential composing reactions. An action is a process that transforms any set of nouns to another.

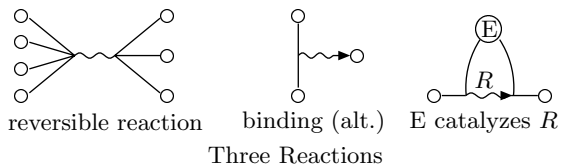
There are three types of actions, each with its own symbol:

*Bindings:* A binding takes two molecules and produces a third, or takes a bound state and break it into two pieces. In the second case, the binding is called an unbinding. The notation for these actions was first introduced by Kohm[1], since this is the principle action in his notation. The bound state is represented by a filled circle, with lines at opposite sides that link it to the two components. The two outgoing lines must extend into each other; the angle between them must be  $\pi$ . This constraint is added to avoid ambiguity in complex binding schemes in which the same dimer may be formed in several different ways.

A binding is usually reversible. If it is not, arrows on the lines indicate the direction.



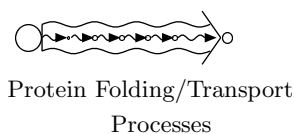
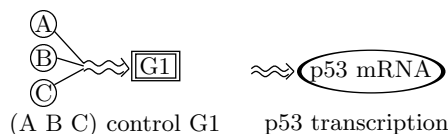
*Reactions:* a reaction, or regular action, consists of a fast joining and splitting of multiple inputs to produce multiple outputs. It is represented by a squiggle joining two vertices linked by lines to the inputs and outputs. An arrow on the squiggle represents an irreversible action. An enzyme catalyzed reaction is a reaction as well, with the enzyme appearing as both reactant and product. A formation reaction is annotated as a squiggle attached at one end to nothing, as is a degradation. Actually, of course, there are always products in a degradation, and there are always ingredients for formation. Often it is not necessary to keep track of them, and this is especially true if they are ubiquies.



Note that a binding is a special case of a reaction, a reaction with two inputs and one output. That is a redundancy in

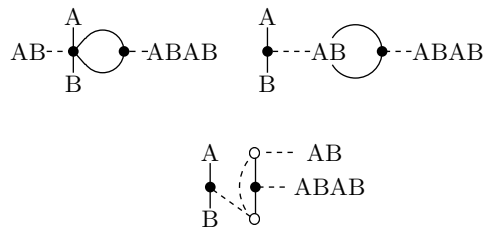
the notation, since a binding may be represented two ways. This is useful, since binding notation is more compact. But, if the modeler wants to refer to the binding reaction squiggle, it should be represented as a 2-1 reaction.

*Processes:* these also take multiple inputs to multiple outputs, but are slow for one reason or another. In order to calculate the production rate of the outputs, the modeler must remember the entire history of the inputs, since inputs that entered the process at a much earlier time might still be present in an intermediate stage. Processes are represented by a double squiggle, with arrows to indicate irreversibility.



The distinction between a process and a reaction is not a sharp one. If an action takes a minute to proceed, but all the inputs vary on the time scale of hours, the process may safely be modeled as an instantaneous reaction, with biologically irrelevant error. If, on the other hand, the process takes a minute, but the reactants vary on a time scale of seconds, it must be modeled as a process.

In order to reduce clutter, it is convenient to annotate different actions in different places. The *equivalence* allows two nouns (atoms, nounboxes, or linkboxes) in the diagram to represent the same entity. Using equivalences, the modeler can describe different actions a particular object participates in without cluttering the diagram with lines. The equivalence is denoted by a dashed line, an *equivalence line*.

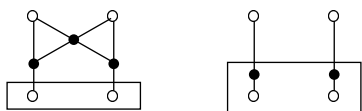


three diagrams that represent the same network

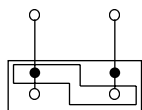
If two nouns are linked by an equivalence line, any actions that one participates in is added to the actions that the other participates in to form a larger list of actions. In this way, the complete list of actions for an object can be listed in separate subdiagrams. In practical applications, we find a small number of diagrams sufficient.

### 3. LINKBOXES AND LIKEBOXES

The first complex grammatical element in the notation is the *linkbox*. The linkbox is a noun that represents the physical joining of one, two, or more nouns, and can be thought of as a generalization of the "combination of states" introduced by Kohn. Placing two objects in a linkbox represents the two things present simultaneously. A protein is a linkbox of its binding sites, DNA is a linkbox of the genes it contains. Objects joined with a linkbox diffuse together, and participate in reactions as a whole. Linkboxes may also refer to subparts of linked objects. In this case, the linkbox represents the state of the subpart of the object. If a linkbox surrounds the bound form of two binding sites in a protein with seven binding sites, the linkbox represents the protein with the two binding sites occupied, no matter what the state of the other binding sites is. Along with the equivalence line, the linkbox allows the modeler to refer to any state of a molecule, in principle.



Two Linked Sites Independent Binding



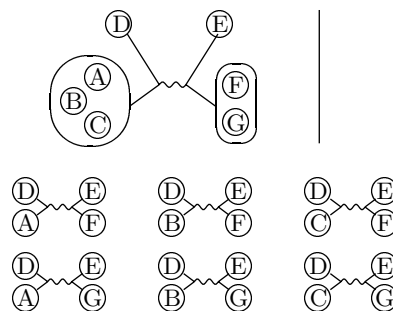
Interior Linkbox

linkboxes of two binding sites representing different states

A linkbox is an object composed of other objects, and acts as a noun. The analogous objects for verbs have already been described— these are the processes. A process is composed of several sub-actions in succession. If a process squiggle surrounds a set of actions, that process is composed of the actions in succession. The process squiggle, unlike the linkbox, is inherently a linearly ordered object, since time (alas) is one-dimensional.

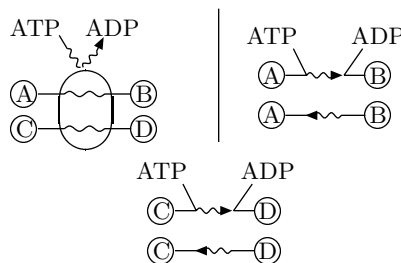
With the notation introduced so far, it is possible to annotate any process or action. The notation is already complete. It is already modular since it includes reactions and processes. The third grammatical element we introduce is therefore unnecessary in principle, but useful in practice.

The *likebox* is the third complex grammatical element, and it is the most complex. The likebox defines sets of objects that act alike. Any elements of the likebox share all the properties of a likebox.



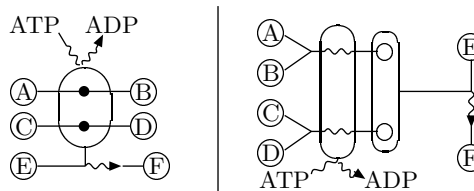
The diagram with likeboxes is equivalent to six diagrams

If a likebox surrounds a set of atoms or linkboxes (or other noun-like likeboxes), it is a *nounbox*. A nounbox acts as a noun. It participates in reactions as an atom does. If the likebox surrounds a set of actions, it is a *verbbox*, in which case the properties of the likebox are interpreted as properties of the actions.



The verbbox is short for the four reactions on the right

In the special case that a likebox surrounds a set of dimers, and no other nouns or verbs, the likebox is interpreted as both a nounbox and a verbbox, depending on the context. If the likebox is an input to a reaction, the likebox is acting as a nounbox of the dimers. If the likebox is enzymatically stimulated, the enzymatic stimulation is applied to the dimerization reactions instead. If the modeler wishes to refer to the verbbox or nounbox separately, an equivalence line can link the ambiguous box to an unambiguous box.



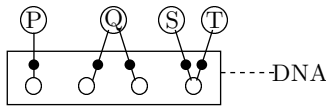
A binding likebox is both a verb-box and a noun-box

likeboxes and linkboxes interact grammatically in complex ways.

### 4. SIMPLE EXAMPLES

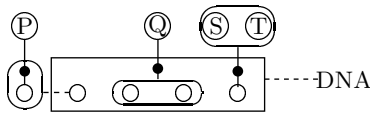
The first example of the notation is a set of simple binding reactions. P,Q,S and T are 4 transcription factors which

bind to four binding sites. P binds to the first of the four binding sites, Q binds to two others, and S and T bind competitively to the last.



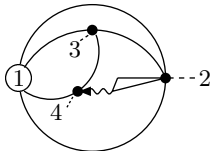
P and Q bind independently of other transcription factors  
S and T bind competitively to one site.

A different form of the same diagram, but using the likebox construction, is the second example. It is clear that the likeboxes allow the annotater to avoid a forest of intersecting lines in the case where many bindings can occur. In this example, one of the binding sites is removed from the linkbox representing the DNA. In order to keep the meaning of the diagram the same, the site must be declared equivalent to a likebox of the unbound and bound form. This implies that the linkbox contains an *implicit likebox*, even when there are no explicit likeboxes present. This is one of the great virtues of the notation, and it will be elaborated on later.



An equivalent diagram

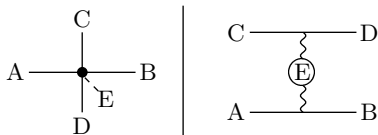
Another example is the binding of 4 subunits to form a functional protein. This occurs in the formation of p53 and hemoglobin. In p53, the case we will consider, the subunits are identical.



The Binding of p53 from four identical monomers. The numbers represent the number of bound monomers in the particular atom

## 5. SHORTHAND NOTATION

One of the notations we have already introduced is a shorthand. The notation for binding encapsulates a reaction with two reactants and one product, or vice-versa.



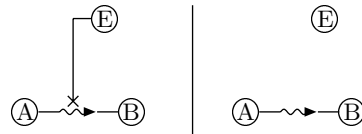
The two forms are equivalent

In a cell, enzymatically catalyzed reactions are so common that it is essential to have a shorthand for them as well. In

an enzyme catalyzed reaction the enzyme appears as both a reactant and a product. Furthermore, the default assumption is that the enzyme is unchanged. This is represented by a line stretching from the enzyme noun to the reaction squiggle. The end of the line has an arrowhead which describes the type of enzymatic action.

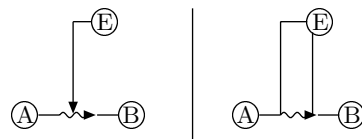
There are four types of enzymatic action:

*Do-Nothing:* this means the enzyme does nothing to the reaction. It is useful when resolving like-boxes to indicate that certain components of a likebox do not act. The arrowhead symbol is a cross.



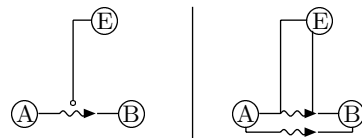
two equivalent diagrams— E does not participate in the reaction

*Catalyze:* The enzyme is required to catalyze the reaction. It cannot proceed at an appreciable rate without it. The arrowhead is an open triangle.



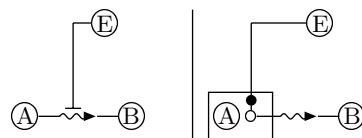
E is required for the reaction to take place

*Promote:* The enzyme promotes the reaction. The reaction proceeds at a slower rate in the absence of the enzyme. This arrowhead is a circle.



Enzymatic promotion

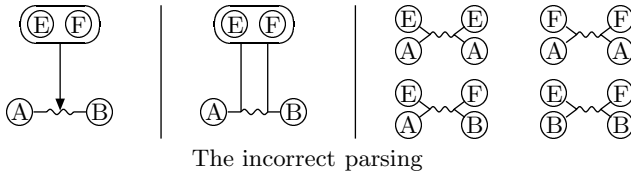
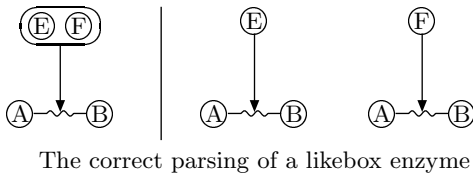
*Inhibit:* The enzyme prevents the reaction from taking place. The arrowhead is a line. Inhibition is an involved process, usually involving competitive binding of the enzyme. In fact, inhibition is a shorthand for some process which prevents the specific reaction from taking place, but does not prevent other reactions from taking place. We shall choose one representative of this class to interpret as inhibition. Any other representative of this class of reactions would do. In the example, we interpret the inhibition to take place through competitive binding to a binding site physically linked to the reactant.



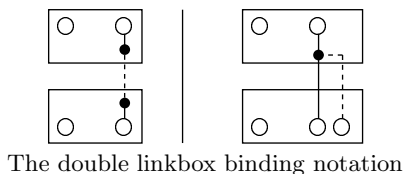
## One interpretation of enzymatic inhibition

If the enzyme is involved in a binding, the arrowhead is pointed toward the filled circle that represents the bound state. This is because there is an implicit reaction squiggle inside the binding circle. If the enzyme points to an verb-box, the enzyme has the same effect on each component.

In all cases, an enzymatic line is the same as two lines from the enzyme, one to each end of the action. Technically, however, the like-boxes make interpretation slightly ambiguous. If a likebox has three elements all of which catalyze a reaction, it is understood that each enzyme catalyzes the reaction separately, not that each enzyme can appear as a reactant and a product. One enzyme cannot enter the reaction only to have another enzyme come out. No human would ever be confused, but a computer could be. The formal resolution of the ambiguity is the precedence rule: the likebox is resolved into three cases before each enzyme line is replaced by two lines to each end of the reaction. The order makes a difference, and this order is the intuitive one.

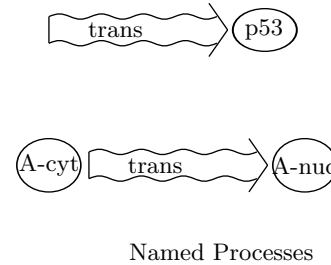


The final shorthand we introduce is a shorthand for a binding represented in two different positions. If a binding site on one protein binds a site on another protein, and the dimer participates in all the reactions of its constituents, then it is convenient to indicate this by placing a node representing the bound sites in each of the two linkboxes that represent the constituent molecules. The shorthand notation is a binding with one end a solid line and the other end an equivalence line, stretching to a mirror-image partner. This is the binding-equivalence object, and it means that there is a bound state of the two linkbox components which belongs to both linkboxes at the same time.



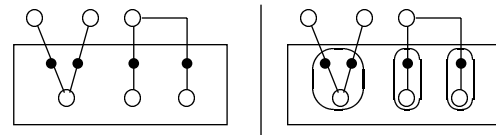
When a process is quite common, examples are translation and transcription, it is convenient to introduce a shorthand for it. This introduces no new grammatical ambiguities. A

formation process with the word "trans" inside could represent the transcription of mRNA, the translocation of the mRNA to the endoplasmic reticulum, the excision of the introns, the translation of the protein, the folding of the protein, and the translocation of the folded protein to the appropriate destination. This is a time-delay module. The transport of a protein from the cytosol to the nucleus in a eukaryotic cell is also a time-delay module and may be denoted by a "trans" labeled arrow as well.



## 6. IMPLICIT LIKEBOXES

A linkbox can contain several instances of its components. If the linkbox represents a protein like p53, with multiple phosphorylation sites, or a stretch of DNA with several promoter binding sites, there is an exponentially large number of states of the same molecule. The notation is simplified by letting a linkbox contain more than one form of its components simultaneously. If a site can be phosphorylated, the phosphorylated form may be placed inside the linkbox. The proper interpretation is that the linkbox contains a linkbox of the two phosphorylation states, in addition to other binding sites.



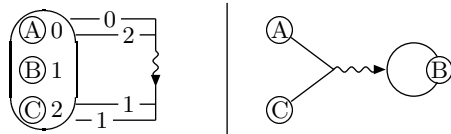
The implicit linkboxes are made explicit in the right-hand picture

This linkbox is implicit, which means that it does not need to be drawn. It is present whenever an action takes place with one input and one output inside a linkbox. If there are other actions that the binding site takes part in, and the output is inside the linkbox, the implicit linkbox surrounds those as well.

The concept of implicit linkboxes reduces the complexity of many diagrams a great deal. If bindings may occur in several different ways, the result is a forest of possibilities indicating each path that the reaction can take. The implicit linkbox allows a modeler to annotate many paths with one process line. This often has the effect of pruning the forest of possibilities which has complicated previous notational efforts.

If the linkbox participates in a reaction, all the different possibilities for the implicit linkboxes participate in the same reaction. If a sub-linkbox contains components of an implicit

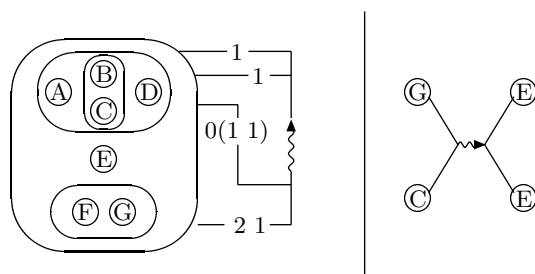
likebox, there is an implicit sub-likebox associated with the linkbox. The linkbox inherits the numbering of its likeboxes, separated by a comma. This is the subject of the next section.



Likebox Numbering

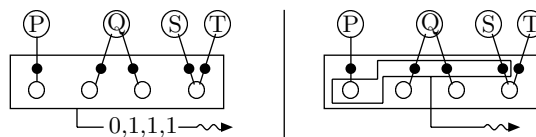
Sometimes we wish to separate out single instances of a likebox for special treatment. Perhaps one of many like enzymes has a different effect on a particular reaction. Perhaps the different phosphorylation states of a complex protein do very different things. In order to allow this, we introduce a numbering scheme to identify the different components of a likebox. Each component is given a number, either explicitly, by placing a number next to it, or implicitly from left to right and top to bottom with the numbers increasing in the same order as an English reader reads. With a nod to computer science, all counting begins at zero. This, of course, is the only correct way to count. There is one exception to this rule— if the likebox is implicit, and one of the cases is known to be the unbound state of a binding site, this site is numbered zero by default. When a line emerges from the likebox, if the line has a number attached, then the line refers only to the case indicated. If the line forks into two lines, and a number is attached to one of the lines, the numbered line refers to its case, and the unnumbered line refers to all the others.

A likebox can contain other likeboxes, in which case, certain numbers refer to likeboxes. When this happens, the inner likebox is unresolved. If the modeler wishes to resolve the inner likebox, the resolution of the interior likebox should appear immediately after the number that indicates the likebox, separated by a space, and enclosed in (optional) parentheses. This is a recursive definition, since likeboxes can nest arbitrarily deeply.



Nested Likebox Numbering Scheme

A linkbox inherits the numbering of the implicit likeboxes it contains. To resolve a linkbox, the likebox resolutions are listed, separated by a comma.



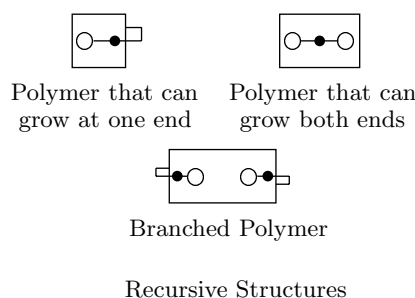
A Linkbox Inherits Implicit Likebox Numbers

When two resolvable objects participate in an action, the resulting output may be different depending on which instances participate. In order to resolve the ambiguity, the output inherits the resolution of all the resolvable participants into a comma separated list in english-language order. If there are several actions that produce the same object, they are lettered A,B,C, explicitly or implicitly in english language order. If there are more than 26 possible actions, we extend the notation in an obvious way— after Z comes AA, then AB, etc. To resolve a node, proceed backwards through its history. First, write down the letter of the reaction that produced the node, then write the resolution of the likeboxes in a comma separated list. If there is a linkbox as one of the reactants, write down the resolution of the linkbox in parentheses, as one of the elements of the list.

## 7. COMPLEX EXAMPLES—RECURSION AND GENERAL COMPUTATION

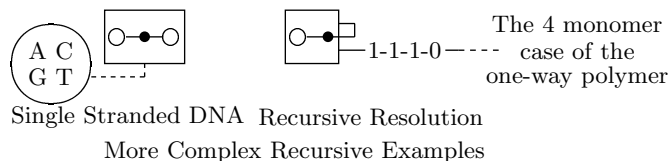
This notation opens the door to dangerous recursion, since the resolution of the history of the object might come back to referring to the same object once again. The clearest example of this is the process of polymerization, since a polymer may join a linkbox of monomers to form the polymer once more. In principle, infinite length histories may occur, depending on how the monomers joined and were removed.

The polymer diagram provides an interesting example of recursive resolution. In this case, the monomer molecule has a binding site, which may bind the molecule itself, whether or not there are monomers already bound. Applying the resolution operators, it is very easy to trace back the history and refer to a particular length of polymer.



A more complicated example is a branch polymer diagram. In this case, there are two separate sites which may bind the molecule as a whole, in either bound or unbound form. In this case, the recursive resolution resolves the branched polymer structure.

Single-stranded DNA itself is a complicated recursive binding of ATGC. The structure of DNA is indicated in the diagram below.



Single Stranded DNA Recursive Resolution  
More Complex Recursive Examples

The recursive properties of biological molecules and their computational structure are made clear by the form of this notation.

## 8. CONCLUSIONS

The notation we describe has two appealing features. The same notation can describe modules and individual reactions, and the notation can be mechanically parsed into a set of individual reactions. The reactions have a rate which may be specified as a function of the concentration of the reactants, and then the diagram immediately translates into a computer model.

This is one of the central features of the notation— it parses into a skeleton model which is fully specified by a list of parameters. In the case where all the actions are elementary, only simple bindings and unbindings, no likeboxes or linkboxes, we may write down a set of differential equations from the diagram. The state of the system is described by the concentrations of the atoms. The rate of change of each concentration is a list of terms. There is one term for each binding that has the atom as an output, and one term with a negative sign for each binding that has the atom as an input. The result is a set of standard mass-action differential equations.

When reactions, likeboxes, linkboxes, and processes appear, the grammar of the language must reduce a complex diagrams to a set of elementary processes. Each process has a set of parameters that describes it. Without specifying the formal rules, the notation can be ambiguous for complex diagrams. But for nearly all diagrams that we have produced, the heuristic rules described herein are sufficient to unambiguously parse the diagrams. We will provide the formal grammar for the language in a separate publication, since it is of interest mostly to specialists, while we believe the notation will be of interest to a wide audience of biologists.

The notation introduced has been far more economical in practice than currently existing notation.

For a large annotated network, we refer the reader to [http://www.gnsbiotech.com/prod\\_docs\\_public/diag.shtml](http://www.gnsbiotech.com/prod_docs_public/diag.shtml).

## 9. ACKNOWLEDGEMENTS

We would like to thank Caroline Combs for putting the notation into practice so effectively. We would also like to thank Basudev Chaudhuri, Andrew Goldenson, Colin Hill, Iya Khalil, and Vipul Periwal, and all our colleagues at Gene Network Sciences for their helpful comments.

## 10. REFERENCES

- [1] K. W. Kohn. Molecular interaction map of the mammalian cell cycle control and dna repair systems.