Bioinformatics Computational & Systems Biology

http://elynsgroup.com

Review Article | Journal of Bioinformatics, Computational and Systems Biology | Open Access

The Cell Embodies Standard Engineering Principles

Curtis E. Hrischuk[1] and Gregory T. Reeves[2]*

[1]*ACM, Morrisville, NC 27709, USA*

[2]*Department of Chemical and Biomolecular Engineering, North Carolina State University, Raleigh, NC 27606, USA*

Received Date: July 01, 2017, **Accepted Date:** July 31, 2017, **Published Date:** August 04, 2017.

***Corresponding author:** Gregory T. Reeves, Department of Chemical and Biomolecular Engineering, North Carolina State University, Raleigh, NC 27606, USA, Tel: 919-513-0652; Fax: 919-515-3465: E-mail: gtreeves@ncsu.edu.*

Abstract

The structural and operational complexity of the cell is confounding. Indeed, the cell has been likened to complex man-made systems, such as a chemical manufacturing plant or a computer. However, the depth of these analogies has not been sufficiently explored from an engineering perspective, nor have the pragmatic implications. In this work, we demonstrate the cell embodies three different sets of previously-published standard engineering principles (SEPs): General Engineering Principles, Hardware/Software Co-design Principles, and Robot Engineering Principles. The latter two are especially relevant, as the cell is both an embedded computing system and an embodied, autonomous agent. We also develop a set of Chemical Process Control Engineering Principles and demonstrate that these apply to the cell as well. Our observations have several implications, including that we should pragmatically expect that any individual biological system under study would exhibit characteristics of an engineered system. This implication can be formulated as a predictive model for discovery, which we call the "Engineering Principle Expectation". This expectation can be used as a criterion for research when one of many paths can be selected. It can also serve as a validation criterion for assessing the strength of research results. Lastly, it points to a need for engineering disciplines to look to biology for expanding the SEPs that can be applied within those disciplines.

Keywords: Systems biology; Embedded system; Process control

Abbreviations

Standard Engineering Principles (SEPs); General Engineering Principles (GEPs); Co-design Engineering Principles (CDEP) and Robot Engineering Principles (REPs); Chemical Process Control Engineering Principles (CPCEP); Engineering Principle Expectation (EPE).

Introduction

In the last several decades, different engineering disciplines have explored ways to harness biology. Engineers have looked to biology not just for medical applications, but also for production of biomimetic materials and technologies, neural network processing algorithms, and even computer networking [1]. These engineering applications of biological phenomena are made possible by the fact that biological systems include sophisticated protein machines, communication protocols, and other features that suggest analogies with engineered systems [2,3]. These cross-domain analyses of engineering and biology have been piecemeal and without a framework for analysis or comparison. What is missing from this synergism is an over-arching principle that unites biology with engineering disciplines. We have previously taken a step in this direction by identifying the cell as an embedded computing system [4]. As such, the models used by the engineering sub-discipline of embedded computer engineering could be applied to systems biology modeling tasks. In this paper, we survey the relevant Standard Engineering Principles (SEPs; otherwise known as "best practices") for engineering embedded systems, and then examine the cell for which of those principles are evident. We find that the cell embodies SEPs from three sub-categories of engineering that each apply to embedded systems: General Engineering Principles (GEPs) [5], Hardware/Software Engineering Co-design Principles [6] (CDPs), and Robot Engineering Principles (REPs) [7]. A fourth engineering sub-category, process control, also applies to embedded computing systems. Here we develop a set of Chemical Process Control Engineering Principles (CPCEP) and show that these also apply to the cell.

In practice, SEPs within each engineering discipline are followed to robustly and efficiently engineer products. In order to perform function "X" efficiently, human engineered systems conform to these SEPs. These SEPs are determined over the years through the painstaking trial-and-error refinement of engineering, manufacturing, and troubleshooting processes – perhaps very similar mechanisms to those by which evolving systems are thought to explore solutions. Therefore, we argue that we should expect the complex systems of the cell to embody such principles as well. This is especially true considering (1) for a cell, failing to perform "X" efficiently compromises fitness, and (2) there is no operator to guide the cell when unexpected disturbances arise.

The presence of engineering principles within the cell implies that SEPs can be used as starting point to formulate hypotheses about how a cell operates and behaves. In other words, we should pragmatically approach the cell as an engineered system and use that point of view to predict (hypothesize) the expected behavior of biological systems. We call this approach the *Engineering Principle Expectation* (EPE).

In the next section, we briefly review engineering principles that would apply to the cell as an embedded computer: there are twelve General Engineering Principles as defined by Berztiss [5], five Hardware-Software Engineering Co-design Principles [6], thirteen Robot Engineering Principles [7], and six Chemical Process Control Engineering Principles (developed here). With these principles in hand, we then conclude with a discussion of the implication of these observations, including the aforementioned "Engineering Principle Expectation".

Standard Engineering Principles

In this section of the paper, we introduce each of the four categories of SEPs and briefly describe how examples of these SEPs apply to the cell. The full list of all four sets of SEPs, and how they each can be applied to the cell, can be found in the Supplementary Information. Most of the SEPs are easy to understand and apply to the cell so readily that only college-level textbook knowledge is required. In other cases, the principle itself, and perhaps the manner in which it applies to biology, is more difficult to understand. These descriptions will have more in-depth explanations.

General Engineering Principles

Moving from a design to an implementation is the engineering

process. Through the repeated exercise of this process, best practices are codified as general engineering principles. As these general engineering principles have been found to apply to building complex systems, it is no surprise that complex biological systems also exhibit those same principles. Berztiss lists twelve General Engineering Principles (GEPs; see supplementary information) [5]. A review of the list makes clear that direct analogies can be drawn between most of the twelve GEPs and specific cellular processes. Here we discuss three of the twelve GEPs identified by Berztiss. We give an exhaustive treatment of all twelve GEPs in the supplementary information.

GEP 1: Development of engineered objects follows a plan in accordance with quantitative requirements. The biological world is replete with examples of structures that require carefully orchestrated processes in their construction. As a textbook example, the process of constructing a cell (i.e. cell division) follows a very intricate plan and must pass through several quantitative checkpoints before completion [8]. Examples could be multiplied.

GEP 2: Requirements are ranked according to cost-effectiveness [resource use and expenditure], and the development plan, which has an incremental structure, emphasizes the higher ranked requirements. Systems are structured and built incrementally with a layered topology, and not all things are prioritized equally. For example, the regulatory systems that control gene expression can be considered to have three hierarchical levels: the sequence level, the chromatin level, and the nuclear level [9]. The energy (ATP) usage of the cell is also organized hierarchically. Finally, one of the most seminal examples of this principle is the diauxic growth that derives from preferential glucose utilization [10].

GEP 3: Standards are used where available and applicable, with every departure from applicable standards explicitly justified. Common design patterns are called standards. For example, screw and nut sizes are defined by standards. Standards are the enabler of common components and/or patterns. Biological systems are ubiquitous with common components and/or patterns. The sets of 20 amino acids, 4 deoxyribonucleotides, and 4 ribonucleotides are the most basic examples of the common parts of biological systems. Such a list could also include lipids, sugars, and other basic molecular components found in the cell. At a high level, transcription, translation, and replication are standard components for all cells. The energy grid of the cell is standardized on the ATP molecule. This extends to a genetic code that is the standard way of encoding the software of the cell.

Departures from these standards are rare. As an example, there are some documented cases of non-canonical amino acids being naturally incorporated into proteins, including the so-called "21st amino acid," selenocysteine. We can speculate as to the reasons behind the departures from the standard. In the case of selenocysteine, it appears the selenium atom has unique, advantageous properties that can be used in special circumstances, such as in antioxidant chemistry [11].

Hardware-Software Engineering Co-Design Principles

In a general-purpose computer system, the hardware and software components are designed and manufactured separately. On the other hand, embedded systems, such as the on-board computer in a car or the flight control computer on a passenger jet, are specifically designed to interface with the environment in to which they are embedded. There are other types of computer systems too. So what type of computer system is comparable to a cell? Reeves and Hrischuk [4] answered that the cell is an embedded system because it exhibits the following properties:

- *Environmental interaction* via multiple types of sensors and control elements (actuators) [12].
- *Concurrency* that handles multiple independent stimuli simultaneously [12].
- *Reactiveness* that responds in a timely manner to avoid system failures.
- *Liveness* so the critical processes must not terminate or stall.
- *Robustness* by adapting to changing conditions, including internal failures. It also functions properly if resources are unavailable for periods of time, when data is delivered with variable delay, or even if data is missing [13].
- *Heterogeneity* with processing capabilities that span various computational styles and implementation technologies.

Because the cell is an embedded system, it also is faced the "hardware-software co-design problem," in which both aspects must be designed together in order to meet functional, performance, cost, and reliability constraints [6].

Wolf identified five co-design engineering principles (CDEP) that are used to solve these problems [6]. Here we present three of the five principles; they are discussed in their entirety in the Supplement.

CDEP 1: Partitioning the function to be implemented into smaller interacting pieces. This is seen in the modularity of both computer hardware and software which have subsystems that work together in their environment to accomplish a function. In a similar way, cellular processes are composed of autonomous acting proteins (processing elements and machines) that join together to accomplish a function, both at the cellular level and at the multicellular organism level. As we have seen, cellular regulatory networks can be decomposed into modules, which are groups of closely interacting network elements that work together to achieve a specific function [14,15]. For example, a set of genes (that act together as a "module") has been identified that acts to specify the endodermal and mesodermal tissues in a developing sea urchin [16]. Similarly, a set of genes act in a module to specify the dorsal-ventral axis in the early *Drosophila melanogaster* embryo [17].

CDEP 2: Allocating those partitions to microprocessors or other hardware units, where the function may be implemented directly in hardware or in software running on a microprocessor. In the human world, the ideal embedded solution optimally would be a separate processor for each function, provided that power and economic budgets allowed. In reality, we are limited by the manufacturing and programming constraints. However, the cell is able to achieve this ideal: each protein or protein complex operates as a separate hardware unit. In the example of the regulatory networks described above, each module is a group of genes and proteins that can act separately from other modules.

At a different level of abstraction, in bacteria, multiple genes that correspond to a single regulatory module are co-regulated in an operon. These genes are found together in the DNA and their transcription occurs at the same time.

CDEP 3: Scheduling the times at which functions are executed. This is important when several functional partitions share one hardware unit. The life of a cell follows the cell cycle, which is a series of tightly regulated process. A control system with several layers adjusts when key components of the cell cycle machinery are produced, activated and degraded to make sure that the schedule is kept [18]. Recently, electrical engineering techniques were used to identify general modules for gene regulation, such as switches or oscillators. This includes feedback loops for

autoregulation where a protein modifies, directly or indirectly, its own rate of production [19].

Unless one recognizes the cell as an embedded system [4], it may be surprising that the cell matches these co-design principles so closely. In other words, the close alignment of these co-design principles with the cell can be considered validation for the hypothesis that the cell is an embedded computer system.

Robot Engineering Principles

Robot engineering principles (REPs) are relatively new. Pfeifer et al. [7] formulated thirteen robot engineering principles that apply to the design robots as embodied, autonomously-acting agents. The authors discuss how the embodiment requires all aspects of the agent (sensors/actuators, circuitry, moving limbs) to be highly connected. As such, it is difficult for the integrated whole of the autonomous robots, control systems and software, manufacturing processes, etc. to leave out any element without the whole system failing. The cell (as well as proteins within the cell; [4]) also qualifies as an autonomously-acting agent that is an integrated whole of many interacting processes. Furthermore, many of the REPs deal with the interaction between the agent and the environment, an obvious property of cells. Therefore, it is unsurprising that many REPs also apply to the cell. Here we discuss how three of these REPs apply to biology (for a full treatment of all 13 REPs, see the Supplementary Material).

REP 1: The synthetic methodology principle. "*Understand by building*". This principle is manifested as the incremental growth of complexity of cellular systems, on a geological time scale. On a more local time scale, random variation is a form of building slightly different structures, while natural selection weeds out the structures that were built, but were detrimental. In this manner, the cell has "understood" what structures work and which ones do not (in a similar way in which we describe, in GEP 11, the adaptation that occurs over generations of biological systems as learning).

REP 2: Emergence. "*Systems should be designed for emergence (for increased adaptivity). This emergence in the behavior of the agent is the result of a system-environment interaction. Systems designed for emergence tend to be more adaptive and robust*".

In this principle, Pfeifer et al. [7] describe emergence as a set of behaviors that arise in the agent that were not initially pre-programmed. It is abundantly clear that biological systems embody this principle, as any manner in which organisms adapt their behavior to their environment can be viewed as emergent. Furthermore, as discussed in GEP 4, the cell embodies the adaptivity property. We have discussed the importance of the cell interacting with its environment elsewhere [4], although it is self-evident.

REP 3: Diversity-compliance. "*Agents must maintain the balance between exploiting the specifics of an ecological niche and exhibiting behavioral diversity*".

Pfeifer et al. [7] explain this principle as a trade-off complying with the environmental surroundings, yet showing diversity in the responses to the environment. It is noted that this principle has also been found in other areas of science, including in evolutionary search algorithms [20,21].

Furthermore, this principle is seen as both fundamental to biological systems, as well as problematic in our understanding of how the two aspects of this principle trade-off. In particular, this question often arises in biology when studying how organisms remain in "stasis" within an environment, yet can also adapt quickly in a changing environment. Stephen Jay Gould described this mode of evolution as "punctuated equilibrium," while others have invoked cryptic genetic variation to explain this phenomenon [22].

Chemical Process Control Engineering Principles

While we were not able to find a written, codified set of Chemical Process Control Engineering Principles (CPCEP), we have compiled a (non-exhaustive) list to determine whether the cell embodies such CPCEP.

Most of these principles are the natural consequence of the first we describe: Trade-offs are a fact of life. In an engineered system, there are often multiple goals, and with almost 100% certainty, perfect optimization of one goal results in the detriment of at least one other goal. For example, the design of smart phones is constantly battling the trade-off between a slim phone and a high capacity battery.

However, this trade-off principle does not apply solely to the goals, or performance objectives, of a system. This principle also applies to the sensitivity of a physical system. If a system is sensitive to certain disturbances, implementing process control to increase robustness with respect to these disturbances inevitably results in sensitivity appearing somewhere else [23–25].

Upon inspection of the CPCEPs we have compiled, it is evident that many cellular systems adhere to these principles. We discuss three here, but the entire list of the principles we have compiled, including a full treatment of how they apply to human engineered systems and are embodied by biological systems, can be found in the Supplementary Material.

CPCEP 1: Trade-offs are a fact of life. "*Any attempt to meet a given performance objective (i.e., goal; see below) will inevitably result in an undesired consequence in another area of performance. Trade-offs must be identified and weighed*".

Every physical system has trade-offs [23-26]. For example, there is a trade-off between vehicle size and fuel efficiency. In the chemical process control industry, there is a common trade-off between the speed of your system (how rapidly it attempts to correct an error from set point) and the possibility of overshoot and oscillations. To overcome oscillations, derivative control could be implemented, but this type of control is susceptible to sensor noise (see below). A time-averaging filter could be implemented to reduce noise, but this comes with the trade-off of a more sluggish system.

In the biological world, trade-offs abound; this is an unsurprising fact given that trade-offs are often consequences of the dynamical equations that describe the system. For example, the same trade-off between speed and stability/robustness discussed above can be found in the cell. A cellular pathway designed to react rapidly and excitably to a stimulus can also be ectopically activated by a noise fluctuation. To combat this, a long-pass filter, in the form of a feed-forward loop [27], could be enacted. However, this comes with a trade-off: the system will now respond more sluggishly.

CPCEP 2: Define your performance objective(s). "*Performance objectives should be defined precisely and, if possible, quantitatively. Common performance objectives include meeting product specifications, minimize time to steady state, minimize overshoot and oscillation, minimize waste production, minimize environmental impact, minimize energy use, maximize production rate*".

Each engineered system is built for a purpose. Defining that purpose is an essential step in programming the process control parameters. For example, consider a 30 L tank in a chemical plant designed to hold chemicals. If the tank's purpose is to hold the chemicals before sending them to a plug flow reactor, then strictly controlling the level in the tank is not as important as the outlet flow rate. If the tank's purpose is to hold a mixture of chemicals for

a given residence time so that a reaction can take place (i.e. if the tank is a continuous stirred tank reactor), then both the level in the tank as well as the flow through the tank are important.

Besides the need to know what types of control to implement, the reality of trade-offs also underscores the need to define performance objective(s) clearly. There are always competing objectives, so choosing which one, or ones, to attempt to satisfy is the important first step.

The situation is the same in biological systems. Each sub-process has a purpose to increase the fitness of the cell, so this can be understood to be the overarching performance objective. However, these sub-processes operate to achieve local objectives as well. Returning to the example of the feedforward loop serving as a long-pass filter [28,29]: such a feedforward loop is detrimental if the objective of the system is to achieve a rapid, excitable response to the initial stimulus. On the other hand, an excitable response can be undesirably activated simply by noise, which may be why the feedforward loop occurs so frequently in nature [27].

CPCEP 3: Do not under- or over-meet objectives. "*Strategies should be implemented to only just meet objective(s), because loose/sluggish control costs money, but over-aggressive control can result in extra cost, instabilities, or failure to meet other objectives. Objectives must be met so that they fall within the standards established by the (quantitatively defined) performance objectives*".

The value of landing within the established, narrow range of allowable values for the product should be self-evident. For example, if the objective is to maintain the chemical purity of a product, then an impure product may not pass regulatory approval, yet an overly pure product does not net any more revenue (but may cost a great deal more to manufacture). Exacting, aggressive control to maintain an overly pure product may result in an unacceptable amount of environmental waste or undesirable byproduct generated in order to over-purify the product. This would be an example of failing to meet a second performance objective (i.e., low waste generation).

Examples of examining biological systems from the standpoint of trade-offs, performance objectives, and process control are sparse. One of the processes studied in this fashion is morphogen-mediated tissue patterning in development, in which a diffusible molecule called a morphogen (which is usually a protein) is present in a spatial concentration gradient [30]. Cells respond to the presence of the morphogen in a concentration-dependent fashion through differential gene expression. In other words, if a cell perceives the morphogen concentration to be above a certain *threshold*, then it will activate gene expression accordingly. In these systems, when the morphogen production rate is perturbed, the locations where morphogen concentration thresholds occur — and thus, the locations of boundaries between cells of different fates — become perturbed. Without a particular negative feedback control, called "self-enhanced ligand degradation", or SELD, in which the morphogen downregulates its own concentration, the perturbations to cell fate boundaries are unacceptably large [24,31,32]. Thus, SELD negative feedback control reduces the sensitivity of cell fate boundaries to changes in morphogen synthesis rate, and stronger negative feedback leads to more robustness. However, too strong of a negative feedback leads to a loss of morphogen dynamic range and requires over-production of morphogen, while too weak of a negative feedback does not reduce sensitivity enough. The amount of negative feedback should be tuned so that it just meets the standard needed for robustness.

Discussion

In this paper, we have identified several sets of Standard Engineering Principles embodied by the cell. As the cell is an embedded computing system [4], we investigated the applicability of principles from four particular engineering sub-disciplines that are each relevant for embedded systems: General Engineering Principles, Hardware/Software Engineering Co-design Principles, Robot Engineering Principles, and Chemical Process Control Principles.

In retrospect, it is no surprise that the cell embodies these engineering principles. Any sufficiently complex human engineered system adheres to engineering principles. By analogy, if biological systems, which are orders of magnitude more complex than human systems, fail to display engineering principles then the resulting drop in fitness could be catastrophic. Thus, engineering principles were likely fixed in the cell early on. Furthermore, when considering how engineering principles were discovered and evolved (from a human perspective), it is likely that biological systems stumbled upon engineering principles in the same way, through a process of tinkering.

Given the general applicability of human-discovered engineering principles to the cellular world, it seems that, at least pragmatically, biological systems should be studied as if they were engineered systems. This observation regarding biology in general leads to an interesting implication: that any sufficiently complex biological system in particular will exhibit engineering principles as well. This implication can be formulated as a predictive principle, or expectation. As mentioned above, we call this prediction the "Engineering Principle Expectation". Under the right circumstances, this expectation can be very specific if the properly analogous engineering discipline is identified. For example, we might expect cell-cell signaling networks to display the same broad engineering principles as human-designed communication systems. This would include biological modules for signal amplification/reduction, noise filtering, and relay mechanisms. If any of these are not currently known in a given biological communication network, our expectation would be that these modules would be present but are not yet discovered. Thus, the expectation would provide a guide of what to look for when faced with many options.

In addition to a predictive principle, the pragmatic implication can also be applied *post hoc* to enhance our understanding of some biological systems. For example, systems such as the human appendix (initially baffling because it appeared to lack function; see CPCEP4 in the supplementary information), make more sense in light of the need for biological systems to conform to engineering principles.

Our observations also promote the synergy between biology and engineering in two ways. First, we expect known human-discovered engineering principles that have yet to be observed in biology (in general) will be found. Second, given that biological systems are far more complex than man-made systems, we predict that biological systems will guide the development of more advanced engineering principles in other disciplines. A simplified version of this mode of synergy can already be found in biomimetics, in which biological structures provide inspiration for human-designed systems. On the other hand, sometimes human-like designs are discovered in biological systems after the fact ("mimetic convergence"), which aligns more with the first mode of synergy. For example, parallels can be drawn between some aspects of the baroreflex in humans and g-suits worn by fighter pilots. These two aspects of biology/engineering synergy together suggest that the disciplines of engineering and biology both stand to benefit from an engineering approach to studying biological systems.

Conclusion

We have surveyed several sets of previously-published and novel Standard Engineering Principles and have found the cell

embodies these principles. We argue from induction that we should therefore expect biological systems under further study to also exhibit engineering principles (the "Engineering Principle Expectation"). This expectation can be applied both in a predictive and *post hoc* manner. While the cell is far more complex than manmade systems, and thus is likely to display even more advanced principles than those described here, at a minimum, the cell meets our current understanding of SEPs. In summary, there is a tight synergy between biological and engineering science.

Acknowledgements

During the writing of this manuscript, GTR was partially supported by NSF grant CBET-1254344.

Conflict of Interest

Authors declared that they have no conflict.

References

1. Dressler F, Akan OB. A survey on bio-inspired networking. Computer Networks. 2010;54(6):881–900.
2. Navlakha S, Bar-Joseph Z. Distributed information processing in biological and computational systems. Communications of the ACM. 2015;58(1):94–102.
3. Glasscock CJ, Lucks JB, DeLisa MP. Engineered Protein Machines: Emergent Tools for Synthetic Biology. Cell Chemical Biology. 2016;23(1):45–56.
4. Reeves GT, Hrischuk CE. Survey of Engineering Models for Systems Biology. Computational Biology Journal. 2016:1–12.
5. Berztiss AT. Engineering principles and software engineering. In Software Engineering Education. Springer. 1992. p. 437–451.
6. Wolf WH. Hardware-software co-design of embedded systems. Proceedings of the IEEE. 1994;82(7):967–989.
7. Pfeifer R, Iida F, Bongard J. New robotics: design principles for intelligent systems. Artif Life. 2005;11(1-2):99-120.
8. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P. Molecular Biology of the Cell, 4th edition. Garland Science. New York. 2002.
9. van Driel R, Fransz PF, Verschure PJ. The eukaryotic genome: a system regulated at different hierarchical levels. J Cell Sci. 2003;116(Pt 20):4067-75.
10. Monod J. The Growth of Bacterial Cultures. Annual Review of Microbiology. 1949;3(1):371–394.
11. Rahmanto AS, Davies MJ. Selenium-containing amino acids as direct and indirect antioxidants. IUBMB Life. 2012;64(11):863-71. doi: 10.1002/iub.1084.
12. Lee EA. Embedded Software. Advances in Computers. 2002;56:55–95.
13. Albertos P, Crespo A, Vallés M, Ripoll I. Embedded Control Systems: Some Issues And Solutions. IFAC Proceedings Volumes. 2005;38(1):203–208.
14. Davidson EH. Emerging properties of animal gene regulatory networks. Nature. 2010;468(7326):911–920. doi: 10.1038/nature09645.
15. Davidson EH, Rast JP, Oliveri P, Ransick A, Calestani C, Yuh CH, et al. A Genomic Regulatory Network for Development. Science. 2002;295(5560):1669-78. doi:10.1126/science.1069883.
16. Levine M, Davidson EH. Gene Regulatory Networks for Development. Small. 2005;102(14):4936–42.
17. Jermusyk AA, Reeves GT. Transcription Factor Networks. Encyclopedia of Cell Biology. 2016;4:63–71.
18. de Lichtenberg U, Jensen LJ, Brunak S, Bork P. Dynamic complex formation during the yeast cell cycle. Science. 2005;307(5710):724-7.
19. Hasty J, McMillen D, Collins JJ. Engineered gene circuits. Nature.2002;420(6912):224–230. doi:10.1038/nature01257.
20. Eiben AE, Schippers CA. On evolutionary exploration and exploitation. Fundamenta Informaticae. 1998;35(1–4):35–50.
21. Črepinšek M, Liu S.H, Mernik M. Exploration and exploitation in evolutionary algorithms. ACM Computing Surveys. 2013;45(3):1–33.
22. Whitacre JM, Atamas SP. Degeneracy allows for both apparent homogeneity and diversification in populations. Biosystems. 2012;110(1):34-42. doi: 10.1016/j.biosystems.2012.08.003.
23. Csete ME, Doyle JC. Reverse engineering of biological complexity. Science. 2002;295(5560):1664-9.
24. Lander AD, Wing-Cheong L, Nie Q, Wan YMF. The Measure of Success: Constraints, Objectives, and Tradeoffs in Morphogen-mediated Patterning. Cold Spring Harbor Perspect Biol. 2009;1(1). doi:10.1101/cshperspect.a002022.
25. Reeves GT, Fraser SE. Biological systems from an engineer's point of view. PLoS Biol. 2009;7(1):e21. doi: 10.1371/journal.pbio.1000021.
26. Carlson JM, Doyle J. Complexity and robustness. Proc Natl Acad Sci U S A, 99 Suppl 1. 2002. p 2538–2545.
27. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: simple building blocks of complex networks. Science. 2002;298(5594):824-7.
28. Shen-Orr SS, Milo R, Mangan S, Alon U. Network motifs in the transcriptional regulation network of Escherichia coli. Nat Genet. 2002;31(1):64-8.
29. Mangan S, Zaslaver A, Alon U. The Coherent Feedforward Loop Serves as a Sign-sensitive Delay Element in Transcription Networks. Journal of Molecular Biology. 2003;334(2):197–204.
30. Wolpert L. Positional information and the spatial pattern of cellular differentiation. J Theor Biol. 1969;25(1):1-47.
31. Eldar A, Rosin D, Shilo BZ, Barkai N. Self-enhanced ligand degradation underlies robustness of morphogen gradients. Dev Cell. 2003;5(4):635-46.
32. Reeves GT, Kalifa R, Klein DE, Lemmon MA, Shvartsman SY. Computational analysis of EGFR inhibition by Argos. Dev Biol. 2005;284(2):523-35.

Corresponding author: Gregory T. Reeves, Department of Chemical and Biomolecular Engineering, North Carolina State University, Raleigh, NC 27606, USA, Tel: 919-513-0652; Fax: 919-515-3465: E-mail: gtreeves@ncsu.edu.

Received Date: July 01, 2017, **Accepted Date:** July 31, 2017, **Published Date:** August 04, 2017.

Citation: Hrischuk CE, Reeves GT (2017) The Cell Embodies Standard Engineering Principles. J Bioinf Com Sys Bio 1(1): 106.