

= Folding @ home =

Folding @ home (FAH or F @ h) is a distributed computing project for disease research that simulates protein folding , computational drug design , and other types of molecular dynamics . The project uses the idle processing resources of thousands of personal computers owned by volunteers who have installed the software on their systems . Its main purpose is to determine the mechanisms of protein folding , which is the process by which proteins reach their final three @-@ dimensional structure , and to examine the causes of protein misfolding . This is of significant academic interest with major implications for medical research into Alzheimer 's disease , Huntington 's disease , and many forms of cancer , among other diseases . To a lesser extent , Folding @ home also tries to predict a protein 's final structure and determine how other molecules may interact with it , which has applications in drug design . Folding @ home is developed and operated by the Pande Laboratory at Stanford University , under the direction of Prof. Vijay Pande , and is shared by various scientific institutions and research laboratories across the world .

The project has pioneered the use of GPUs , PlayStation 3s , Message Passing Interface (used for computing on multi @-@ core processors) , and some Sony Xperia smartphones for distributed computing and scientific research . The project uses statistical simulation methodology that is a paradigm shift from traditional computational approaches . As part of the client @-@ server network architecture , the volunteered machines each receive pieces of a simulation (work units) , complete them , and return them to the project 's database servers , where the units are compiled into an overall simulation . Volunteers can track their contributions on the Folding @ home website , which makes volunteers ' participation competitive and encourages long @-@ term involvement .

Folding @ home is one of the world 's fastest computing systems , with an approximate speed of nearly 100 petaFLOPS . This performance from its large @-@ scale computing network has allowed researchers to run computationally expensive atomic @-@ level simulations of protein folding thousands of times longer than formerly achieved . Since its launch on October 1 , 2000 , the Pande Lab has produced 129 scientific research papers as a direct result of Folding @ home . Results from the project 's simulations agree favorably with experiments .

= = Project significance = =

Proteins are an essential component to many biological functions and participate in virtually all processes within biological cells . They often act as enzymes , performing biochemical reactions including cell signaling , molecular transportation , and cellular regulation . As structural elements , some proteins act as a type of skeleton for cells , and as antibodies , while other proteins participate in the immune system . Before a protein can take on these roles , it must fold into a functional three @-@ dimensional structure , a process that often occurs spontaneously and is dependent on interactions within its amino acid sequence and interactions of the amino acids with their surroundings . Protein folding is driven by the search to find the most energetically favorable conformation of the protein , i.e. its native state . Thus , understanding protein folding is critical to understanding what a protein does and how it works , and is considered a " holy grail " of computational biology . Despite folding occurring within a crowded cellular environment , it typically proceeds smoothly . However , due to a protein 's chemical properties or other factors , proteins may misfold ? that is , fold down the wrong pathway and end up misshapen . Unless cellular mechanisms are capable of destroying or refolding such misfolded proteins , they can subsequently aggregate and cause a variety of debilitating diseases . Laboratory experiments studying these processes can be limited in scope and atomic detail , leading scientists to use physics @-@ based computational models that , when complementing experiments , seek to provide a more complete picture of protein folding , misfolding , and aggregation .

Due to the complexity of proteins ' conformation space ? the set of possible shapes a protein can take ? and limitations in computational power , all @-@ atom molecular dynamics simulations have been severely limited in the timescales which they can study . While most proteins typically fold in the order of milliseconds , before 2010 simulations could only reach nanosecond to microsecond

timescales . General @-@ purpose supercomputers have been used to simulate protein folding , but such systems are intrinsically expensive and typically shared among many research groups . Additionally , because the computations in kinetic models are serial in nature , strong scaling of traditional molecular simulations to these architectures is exceptionally difficult . Moreover , as protein folding is a stochastic process and can statistically vary over time , it is computationally challenging to use long simulations for comprehensive views of the folding process .

Protein folding does not occur in a single step . Instead , proteins spend the majority of their folding time ? nearly 96 % in some cases ? " waiting " in various intermediate conformational states , each a local thermodynamic free energy minimum in the protein 's energy landscape . Through a process known as adaptive sampling , these conformations are used by Folding @ home as starting points for a set of simulation trajectories . As the simulations discover more conformations , the trajectories are restarted from them , and a Markov state model (MSM) is gradually created from this cyclic process . MSMs are discrete @-@ time master equation models which describe a biomolecule 's conformational and energy landscape as a set of distinct structures and the short transitions between them . The adaptive sampling Markov state model approach significantly increases the efficiency of simulation as it avoids computation inside the local energy minimum itself , and is amenable to distributed computing (including on GPUGRID) as it allows for the statistical aggregation of short , independent simulation trajectories . The amount of time it takes to construct a Markov state model is inversely proportional to the number of parallel simulations run , i.e. the number of processors available . In other words , it achieves linear parallelization , leading to an approximately four orders of magnitude reduction in overall serial calculation time . A completed MSM may contain tens of thousands of sample states from the protein 's phase space (all the conformations a protein can take on) and the transitions between them . The model illustrates folding events and pathways (i.e. routes) and researchers can later use kinetic clustering to view a coarse @-@ grained representation of the otherwise highly detailed model . They can use these MSMs to reveal how proteins misfold and to quantitatively compare simulations with experiments .

Between 2000 and 2010 , the length of the proteins Folding @ home has studied have increased by a factor of four , while its timescales for protein folding simulations have increased by six orders of magnitude . In 2002 , Folding @ home used Markov state models to complete approximately a million CPU days of simulations over the span of several months , and in 2011 , MSMs parallelized another simulation that required an aggregate 10 million CPU hours of computing . In January 2010 , Folding @ home used MSMs to simulate the dynamics of the slow @-@ folding 32 @-@ residue NTL9 protein out to 1 @.@ 52 milliseconds , a timescale consistent with experimental folding rate predictions but a thousand times longer than formerly achieved . The model consisted of many individual trajectories , each two orders of magnitude shorter , and provided an unprecedented level of detail into the protein 's energy landscape . In 2010 , Folding @ home researcher Gregory Bowman was awarded the Thomas Kuhn Paradigm Shift Award from the American Chemical Society for the development of the open @-@ source MSMBuilder software and for attaining quantitative agreement between theory and experiment . For his work , Pande was awarded the 2012 Michael and Kate Bárány Award for Young Investigators for " developing field @-@ defining and field @-@ changing computational methods to produce leading theoretical models for protein and RNA folding " as well as the 2006 Irving Sigal Young Investigator Award for his simulation results which " have stimulated a re @-@ examination of the meaning of both ensemble and single @-@ molecule measurements , making Dr. Pande 's efforts pioneering contributions to simulation methodology . "

= = Biomedical research = =

Protein misfolding can result in a variety of diseases including Alzheimer 's disease , cancer , Creutzfeldt ? Jakob disease , cystic fibrosis , Huntington 's disease , sickle @-@ cell anemia , and type II diabetes . Cellular infection by viruses such as HIV and influenza also involve folding events on cell membranes . Once protein misfolding is better understood , therapies can be developed that augment cells ' natural ability to regulate protein folding . Such therapies include the use of

engineered molecules to alter the production of a certain protein , help destroy a misfolded protein , or assist in the folding process . The combination of computational molecular modeling and experimental analysis has the possibility of fundamentally shaping the future of molecular medicine and the rational design of therapeutics , such as expediting and lowering the costs of drug discovery . The goal of the first five years of Folding @ home was to make advances in understanding folding , while the current goal is to understand misfolding and related disease , especially Alzheimer 's disease .

The simulations run on Folding @ home are used in conjunction with laboratory experiments , but researchers can use them to study how folding in vitro differs from folding in native cellular environments . This is advantageous in studying aspects of folding , misfolding , and their relationships to disease that are difficult to observe experimentally . For example , in 2011 Folding @ home simulated protein folding inside a ribosomal exit tunnel , to help scientists better understand how natural confinement and crowding might influence the folding process . Furthermore , scientists typically employ chemical denaturants to unfold proteins from their stable native state . It is not generally known how the denaturant affects the protein 's refolding , and it is difficult to experimentally determine if these denatured states contain residual structures which may influence folding behavior . In 2010 , Folding @ home used GPUs to simulate the unfolded states of Protein L , and predicted its collapse rate in strong agreement with experimental results .

The Pande Lab is part of Stanford University , a non @-@ profit entity , and does not sell the results generated by Folding @ home . The large data sets from the project are freely available for other researchers to use upon request and some can be accessed from the Folding @ home website . The Pande lab has collaborated with other molecular dynamics systems such as the Blue Gene supercomputer , and they share Folding @ home 's key software with other researchers , so that the algorithms which benefited Folding @ home may aid other scientific areas . In 2011 , they released the open @-@ source Copernicus software , which is based on Folding @ home 's MSM and other parallelization techniques and aims to improve the efficiency and scaling of molecular simulations on large computer clusters or supercomputers . Summaries of all scientific findings from Folding @ home are posted on the Folding @ home website after publication .

= = = Alzheimer 's disease = = =

Alzheimer 's disease is an incurable neurodegenerative disease which most often affects the elderly and accounts for more than half of all cases of dementia . Its exact cause remains unknown , but the disease is identified as a protein misfolding disease . Alzheimer 's is associated with toxic aggregations of the amyloid beta (A β) peptide , caused by A β misfolding and clumping together with other A β peptides . These A β aggregates then grow into significantly larger senile plaques , a pathological marker of Alzheimer 's disease . Due to the heterogeneous nature of these aggregates , experimental techniques such as X @-@ ray crystallography and NMR have had difficulty characterizing their structures . Moreover , atomic simulations of A β aggregation are extremely computationally demanding due to their size and complexity .

Preventing A β aggregation is a promising approach to the development of therapeutic drugs for Alzheimer 's disease , according to Drs. Naeem and Fazili in a literature review article . In 2008 , Folding @ home simulated the dynamics of A β aggregation in atomic detail over timescales of the order of tens of seconds . Prior studies were only able to simulate about 10 microseconds . Folding @ home was able to simulate A β folding for six orders of magnitude longer than formerly possible . Researchers used the results of this study to identify a beta hairpin that was a major source of molecular interactions within the structure . The study helped prepare the Pande lab for future aggregation studies and for further research to find a small peptide which may stabilize the aggregation process .

In December 2008 , Folding @ home found several small drug candidates which appear to inhibit the toxicity of A β aggregates . In 2010 , in close cooperation with the Center for Protein Folding Machinery , these drug leads began to be tested on biological tissue . In 2011 , Folding @ home completed simulations of several mutations of A β that appear to stabilize the aggregate formation ,

which could aid in the development of therapeutic drug approaches to the disease and greatly assist with experimental NMR spectroscopy studies of A β oligomers . Later that year , Folding @ home began simulations of various A β fragments to determine how various natural enzymes affect the structure and folding of A β .

== Huntington 's disease ==

Huntington 's disease is a neurodegenerative genetic disorder that is associated with protein misfolding and aggregation . Excessive repeats of the glutamine amino acid at the N -terminus of the Huntingtin protein cause aggregation , and although the behavior of the repeats is not completely understood , it does lead to the cognitive decline associated with the disease . As with other aggregates , there is difficulty in experimentally determining its structure . Scientists are using Folding @ home to study the structure of the Huntingtin protein aggregate and to predict how it forms , assisting with rational drug design approaches to stop the aggregate formation . The N17 fragment of the Huntingtin protein accelerates this aggregation , and while there have been several mechanisms proposed , its exact role in this process remains largely unknown . Folding @ home has simulated this and other fragments to clarify their roles in the disease . Since 2008 , its drug design approaches for Alzheimer 's disease have been applied to Huntington 's .

== Cancer ==

More than half of all known cancers involve mutations of p53 , a tumor suppressor protein present in every cell which regulates the cell cycle and signals for cell death in the event of damage to DNA . Specific mutations in p53 can disrupt these functions , allowing an abnormal cell to continue growing unchecked , resulting in the development of tumors . Analysis of these mutations helps explain the root causes of p53 -related cancers . In 2004 , Folding @ home was used to perform the first molecular dynamics study of the refolding of p53 's protein dimer in an all -atom simulation of water . The simulation 's results agreed with experimental observations and gave insights into the refolding of the dimer that were formerly unobtainable . This was the first peer reviewed publication on cancer from a distributed computing project . The following year , Folding @ home powered a new method to identify the amino acids crucial for the stability of a given protein , which was then used to study mutations of p53 . The method was reasonably successful in identifying cancer -promoting mutations and determined the effects of specific mutations which could not otherwise be measured experimentally .

Folding @ home is also used to study protein chaperones , heat shock proteins which play essential roles in cell survival by assisting with the folding of other proteins in the crowded and chemically stressful environment within a cell . Rapidly growing cancer cells rely on specific chaperones , and some chaperones play key roles in chemotherapy resistance . Inhibitions to these specific chaperones are seen as potential modes of action for efficient chemotherapy drugs or for reducing the spread of cancer . Using Folding @ home and working closely with the Center for Protein Folding Machinery , the Pande lab hopes to find a drug which inhibits those chaperones involved in cancerous cells . Researchers are also using Folding @ home to study other molecules related to cancer , such as the enzyme Src kinase and certain forms of the engrailed homeodomain ? a large protein which may be involved in many diseases , including cancer . In 2011 , Folding @ home began simulations of the dynamics of the small knottin protein EETI , which can identify carcinomas in imaging scans by binding to surface receptors of cancer cells .

Interleukin 2 (IL -2) is a protein that helps T cells of the immune system attack pathogens and tumors . Unfortunately , its use as a cancer treatment is restricted due to serious side effects such as pulmonary edema . IL -2 binds to these pulmonary cells differently than it does to T cells , so IL -2 research involves understanding the differences between these binding mechanisms . In 2012 , Folding @ home assisted with the discovery of a form of IL -2 which is three hundred times more effective in its immune system role but carries fewer side effects . In experiments , this altered form significantly outperformed natural IL -2 in impeding tumor growth . Pharmaceutical

companies have expressed interest in the mutant molecule , and the National Institutes of Health are testing it against a large variety of tumor models in the hopes of accelerating its development as a therapeutic .

=== Osteogenesis imperfecta ===

Osteogenesis imperfecta , known as brittle bone disease , is an incurable genetic bone disorder which can be lethal . Those with the disease are unable to make functional connective bone tissue . This is most commonly due to a mutation in Type I collagen , which fulfills a variety of structural roles and is the most abundant protein in mammals . The mutation causes a deformation in collagen 's triple helix structure , which if not naturally destroyed , leads to abnormal and weakened bone tissue . In 2005 , Folding @ home tested a new quantum mechanical method that improved upon prior simulation methods , and which may be useful for future computational studies of collagen . Although researchers have used Folding @ home to study collagen folding and misfolding , the interest stands as a pilot project compared to Alzheimer 's and Huntington 's research .

=== Viruses ===

Folding @ home is assisting in research towards preventing certain viruses such as influenza and HIV from recognizing and entering biological cells . In 2011 , Folding @ home began simulations of the dynamics of the enzyme RNase H , a key component of HIV , in the hopes of designing drugs to deactivate it . Folding @ home has also been used to study membrane fusion , an essential event for viral infection and a wide range of biological functions . This fusion involves conformational changes of viral fusion proteins and protein docking , but the exact molecular mechanisms behind fusion remain largely unknown . Fusion events may consist of over a half million atoms interacting for hundreds of microseconds . This complexity limits typical computer simulations to about ten thousand atoms over tens of nanoseconds : a difference of several orders of magnitude . The development of models to predict the mechanisms of membrane fusion will assist in the scientific understanding of how to target the process with antiviral drugs . In 2006 , scientists applied Markov state models and the Folding @ home network to discover two pathways for fusion and gain other mechanistic insights .

Following detailed simulations from Folding @ home of small cells known as vesicles , in 2007 the Pande lab introduced a new computational technique for measuring the topology of its structural changes during fusion . In 2009 , researchers used Folding @ home to study mutations of influenza hemagglutinin , a protein that attaches a virus to its host cell and assists with viral entry . Mutations to hemagglutinin affect how well the protein binds to a host 's cell surface receptor molecules , which determines how infective the virus strain is to the host organism . Knowledge of the effects of hemagglutinin mutations assists in the development of antiviral drugs . As of 2012 , Folding @ home continues to simulate the folding and interactions of hemagglutinin , complementing experimental studies at the University of Virginia .

=== Drug design ===

Drugs function by binding to specific locations on target molecules and causing a certain desired change , such as disabling the target or causing a conformational change . Ideally , a drug should act very specifically and bind only to its target without interfering with other biological functions . However , it is difficult to precisely determine where and how tightly two molecules will bind . Due to limitations in computational power , current in silico approaches usually have to trade speed for accuracy ; e.g. use rapid protein docking methods instead of computationally expensive free energy calculations . Folding @ home 's computational performance allows researchers to use both techniques , and evaluate their efficiency and reliability . Computer -assisted drug design has the potential to expedite and lower the costs of drug discovery . In 2010 , Folding @ home used

MSMs and free energy calculations to predict the native state of the villin protein to within 1 @. @ 8 Å RMSD (root mean square deviation) from the crystalline structure experimentally determined through X @-@ ray crystallography . This accuracy has implications to future protein structure prediction approaches , including for intrinsically unstructured proteins . Scientists have used Folding @ home to research drug resistance by studying vancomycin , an antibiotic of " last resort " , and beta @-@ lactamase , a protein that can break down antibiotics like penicillin .

Chemical activity occurs along a protein 's active site . Traditional drug design approaches involve tightly binding to this site and blocking its activity , under the assumption that the target protein exists in a single rigid structure . However , this approach only works for approximately 15 % of all proteins . Proteins contain allosteric sites which , when bound to by small molecules , can alter a protein 's conformation and ultimately affect the protein 's activity . These sites are attractive drug targets , but locating them is very computationally expensive . In 2012 , Folding @ home and MSMs were used to identify allosteric sites in three medically relevant proteins : beta @-@ lactamase , interleukin @-@ 2 , and RNase H.

Approximately half of all known antibiotics interfere with the workings of a bacteria 's ribosome , a large and complex biochemical machine that performs protein biosynthesis by translating messenger RNA into proteins . Macrolide antibiotics clog the ribosome 's exit tunnel , preventing synthesis of essential bacterial proteins . In 2007 , the Pande lab received a grant to study and design new antibiotics . In 2008 , they used Folding @ home to study the interior of this tunnel and how specific molecules may affect it . The full structure of the ribosome has only been recently determined , and Folding @ home has also simulated ribosomal proteins , as many of their functions remain largely unknown .

= = Participation = =

In addition to reporting active processors , Folding @ home determines its computing performance as measured in floating point operations per second (FLOPS) based on the actual execution time of its calculations . Originally this was reported as native FLOPS : the raw performance from each given type of processing hardware . In March 2009 Folding @ home began reporting the performance in native and x86 FLOPS , the latter being an estimation of how many FLOPS the calculation would take on a standard x86 CPU architecture , which is commonly used as a performance reference . Specialized hardware such as GPUs can efficiently perform certain complex functions in a single floating point operation which would otherwise require multiple operations on the x86 architecture . The x86 measurement attempts to even out these hardware differences . Despite conservative conversions , the GPU clients ' x86 FLOPS are consistently greater than their native FLOPS and comprise a large majority of Folding @ home 's measured computing performance .

In 2007 , Guinness World Records recognized Folding @ home as the most powerful distributed computing network . As of September 30 , 2014 , the project has 107 @, @ 708 active CPU cores and 63 @, @ 977 active GPUs for a total of 40 @. @ 190 x86 petaFLOPS (19 @. @ 282 native petaFLOPS) . At the same time , the combined efforts of all distributed computing projects under BOINC totals 7 @. @ 924 petaFLOPS . In November 2012 Folding @ home updated its accounting of FLOPS , especially for GPUs , and now reports the number of active processor cores as well as physical processors . Using the Markov state model approach , Folding @ home achieves strong scaling across its user base and gains a linear speedup for every additional processor . This network allows Folding @ home to do work that was formerly computationally impractical .

In March 2002 Google co @-@ founder Sergey Brin launched Google Compute as an add @-@ on for the Google Toolbar . Although limited in functionality and scope , it increased participation in Folding @ home from 10 @, @ 000 , up to about 30 @, @ 000 active CPUs . The program ended in October 2005 in favor of the official Folding @ home clients , and is no longer available for the Toolbar . Folding @ home also gained participants from Genome @ home , another distributed computing project from the Pande lab and a sister project to Folding @ home . The goal of Genome @ home was protein design and associated applications . Following its official conclusion in March

2004 , users were asked to donate computing power to Folding @ home instead .

== Performance ==

On September 16 , 2007 , due in large part to the participation of PS3s , the Folding @ home project officially attained a sustained performance level higher than one native petaFLOPS , becoming the first computing system of any kind to do so . Top500 's fastest supercomputer at the time was BlueGene / L , at 0 .@ 280 petaFLOPS . The following year , on May 7 , 2008 , the project attained a sustained performance level higher than two native petaFLOPS , followed by the three and four native petaFLOPS milestones on August 20 and September 28 , 2008 respectively . On February 18 , 2009 , Folding @ home achieved five native petaFLOPS , and was the first computing project to meet these five levels . In comparison , November 2008 's fastest supercomputer was IBM 's Roadrunner at 1 .@ 105 petaFLOPS . On November 10 , 2011 , Folding @ home 's performance exceeded six native petaFLOPS with the equivalent of nearly eight x86 petaFLOPS . In mid @-@ May 2013 , Folding @ home attained over seven native petaFLOPS , with the equivalent of 14 .@ 87 x86 petaFLOPS . It then reached eight native petaFLOPS on June 21 , followed by nine on September 9 of that year , with 17 .@ 9 x86 petaFLOPS . Folding @ home reached 100 x86 petaFLOPS in May 2016 .

== Points ==

Similarly to other distributed computing projects , Folding @ home quantitatively assesses user computing contributions to the project through a credit system . All units from a given protein project have uniform base credit , which is determined by benchmarking one or more work units from that project on an official reference machine before the project is released . Each user receives these base points for completing every work unit , though through the use of a passkey they can receive additional bonus points for reliably and rapidly completing units which are more computationally demanding or have a greater scientific priority . Users may also receive credit for their work by clients on multiple machines . This point system attempts to align awarded credit with the value of the scientific results .

Users can register their contributions under a team , which combine the points of all their members . A user can start their own team , or they can join an existing team . In some cases , a team may have their own community @-@ driven sources of help or recruitment such as an Internet forum . The points can foster friendly competition between individuals and teams to compute the most for the project , which can benefit the folding community and accelerate scientific research . Individual and team statistics are posted on the Folding @ home website .

== Software ==

Folding @ home software at the user 's end involves three primary components : work units , cores , and a client .

== Work units ==

A work unit is the protein data that the client is asked to process . Work units are a fraction of the simulation between the states in a Markov state model . After the work unit has been downloaded and completely processed by a volunteer 's computer , it is returned to Folding @ home servers , which then award the volunteer the credit points . This cycle repeats automatically . All work units have associated deadlines , and if this deadline is exceeded , the user may not get credit and the unit will be automatically reissued to another participant . As protein folding is serial in nature and many work units are generated from their predecessors , this allows the overall simulation process to proceed normally if a work unit is not returned after a reasonable period of time . Due to these deadlines , the minimum system requirement for Folding @ home is a Pentium 3 450 MHz CPU with

Streaming SIMD Extensions (SSE) . However , work units for high @-@ performance clients have a much shorter deadline than those for the uniprocessor client , as a major part of the scientific benefit is dependent on rapidly completing simulations .

Before public release , work units go through several quality assurance steps to keep problematic ones from becoming fully available . These testing stages include internal , beta , and advanced , before a final full release across Folding @ home . Folding @ home 's work units are normally processed only once , except in the rare event that errors occur during processing . If this occurs for three different users , the unit is automatically pulled from distribution . The Folding @ home support forum can be used to differentiate between issues arising from problematic hardware and bad work units .

== = Cores == =

Specialized molecular dynamics programs , referred to as " FahCores " and often abbreviated " cores " , perform the calculations on the work unit as a background process . A large majority of Folding @ home 's cores are based on GROMACS , one of the fastest and most popular molecular dynamics software packages , which largely consists of manually optimized assembly language code and hardware optimizations . Although GROMACS is open @-@ source software and there is a cooperative effort between the Pande lab and GROMACS developers , Folding @ home uses a closed @-@ source license to help ensure data validity . Less active cores include ProtoMol and SHARPEN . Folding @ home has used AMBER , CPMD , Desmond , and TINKER , but these have since been retired and are no longer in active service . Some of these cores perform explicit solvation calculations in which the surrounding solvent (usually water) is modeled atom @-@ by @-@ atom ; while others perform implicit solvation methods , where the solvent is treated as a mathematical continuum . The core is separate from the client to enable the scientific methods to be updated automatically without requiring a client update . The cores periodically create calculation checkpoints so that if they are interrupted they can resume work from that point upon startup .

== = Client == =

A Folding @ home participant installs a client program on their personal computer . The user interacts with the client , which manages the other software components in the background . Through the client , the user may pause the folding process , open an event log , check the work progress , or view personal statistics . The computer clients run continuously in the background at an extremely low priority , using idle processing power so that normal computer usage is unaffected . The maximum CPU usage can be adjusted through client settings . The client connects to a Folding @ home server and retrieves a work unit and may also download the appropriate core for the client 's settings , operating system , and the underlying hardware architecture . After processing , the work unit is returned to the Folding @ home servers . Computer clients are tailored to uniprocessor and multi @-@ core processors systems , as well as graphics processing units . The diversity and power of each hardware architecture provides Folding @ home with the ability to efficiently complete many types of simulations in a timely manner (in a few weeks or months rather than years) , which is of significant scientific value . Together , these clients allow researchers to study biomedical questions formerly considered impractical to tackle computationally .

Professional software developers are responsible for most of Folding @ home 's code , both for the client and server @-@ side . The development team includes programmers from Nvidia , ATI , Sony , and Cauldron Development . Clients can be downloaded only from the official Folding @ home website or its commercial partners , and will only interact with Folding @ home computer files . They will upload and download data with Stanford 's Folding @ home data servers (over port 8080 , with 80 as an alternative) , and the communication is verified using 2048 @-@ bit digital signatures . While the client 's GUI is open @-@ source , the client is proprietary software citing security and scientific integrity as the reasons .

Folding @ home uses the Cosm software libraries for networking . Folding @ home was launched

on October 1, 2000, and was the first distributed computing project aimed at bio @-@ molecular systems. Its first client was a screensaver, which would run while the computer was not otherwise in use. In 2004, the Pande lab collaborated with David Anderson to test a supplemental client on the open @-@ source BOINC framework. This client was released to closed beta in April 2005; however, the approach became unworkable and was shelved in June 2006.

== == Graphics processing units == ==

The specialized hardware of GPUs is designed to accelerate rendering of 3 @-@ D graphics applications such as video games and can significantly outperform CPUs for certain types of calculations. GPUs are one of the most powerful and rapidly growing computational platforms and many scientists and researchers are pursuing general purpose GPU (GPGPU) computing. However, GPU hardware is difficult to use for non @-@ graphics tasks and usually requires significant algorithm restructuring and an advanced understanding of the underlying architecture. Such customization is challenging, especially to researchers with limited software development resources. Folding @ home uses the open source OpenMM library, which uses a bridge design pattern with two application programming interface (API) levels to interface molecular simulation software to an underlying hardware architecture. With the addition of hardware optimizations, OpenMM @-@ based GPU simulations do not require significant modification but achieve performance nearly equal to hand @-@ tuned GPU code, and greatly outperform CPU implementations.

Before 2010 the computational reliability of GPGPU consumer @-@ grade hardware was largely unknown, and circumstantial evidence related to the lack of built @-@ in error detection and correction in GPU memory raised reliability concerns. In the first large @-@ scale test of GPU scientific accuracy, a 2010 study of over 20 @, @ 000 hosts on the Folding @ home network detected soft errors in the memory subsystems of two @-@ thirds of the tested GPUs. These errors strongly correlated to board architecture, though the study concluded that reliable GPU computing was very feasible as long as attention is paid to the hardware characteristics, such as software @-@ side error detection.

The first generation of Folding @ home 's GPU client (GPU1) was released to the public on October 2, 2006, delivering a 20 @-@ 30X speedup for certain calculations over its CPU @-@ based GROMACS counterparts. It was the first time GPUs had been used for either distributed computing or major molecular dynamics calculations. GPU1 gave researchers significant knowledge and experience with the development of GPGPU software, but in response to scientific inaccuracies with DirectX, on April 10, 2008 it was succeeded by GPU2, the second generation of the client. Following the introduction of GPU2, GPU1 was officially retired on June 6. Compared to GPU1, GPU2 was more scientifically reliable and productive, ran on ATI and CUDA @-@ enabled Nvidia GPUs, and supported more advanced algorithms, larger proteins, and real @-@ time visualization of the protein simulation. Following this, the third generation of Folding @ home 's GPU client (GPU3) was released on May 25, 2010. While backwards compatible with GPU2, GPU3 was more stable, efficient and had greater flexibility in its scientific capabilities, and used OpenMM on top of an OpenCL framework. Although these GPU3 clients did not natively support the Linux and OS @-@ X operating systems, Linux users with Nvidia graphics cards were able to run them through the Wine software application. GPUs remain Folding @ home 's most powerful platform in terms of FLOPS; as of November 2012 GPU clients account for 87 % of the entire project 's x86 FLOPS throughput.

Native support for Nvidia and AMD graphics cards under Linux was introduced with FahCore 17, which uses OpenCL rather than CUDA.

== == PlayStation 3 == ==

From March 2007 until November 2012, Folding @ home took advantage of the computing power of PlayStation 3s. At the time of its inception, its main streaming Cell processor delivered a 20x

speed increase over PCs for certain calculations , processing power which could not be found on other systems such as the Xbox 360 . The PS3 's high speed and efficiency introduced other opportunities for worthwhile optimizations according to Amdahl 's law , and significantly changed the tradeoff between computational efficiency and overall accuracy , allowing for the use of more complex molecular models at little additional computational cost . This allowed Folding @ home to run biomedical calculations that would have been otherwise computationally infeasible .

The PS3 client was developed in a collaborative effort between Sony and the Pande lab and was first released as a standalone client on March 23 , 2007 . Its release made Folding @ home the first distributed computing project to use PS3s . On September 18 of the following year , the PS3 client became a channel of Life with PlayStation on its launch . In terms of the types of calculations it can perform , at the time of its introduction the client took the middle ground between a CPU 's flexibility and a GPU 's speed . However , unlike CPUs and GPUs , users were unable to perform other activities on their PS3 while running Folding @ home . The PS3 's uniform console environment made support easier and made Folding @ home more user friendly . The PS3 also has the ability to stream data quickly to its GPU , which was used for real @-@ time atomic @-@ level visualizations of the current protein dynamics .

On November 6 , 2012 , Sony concluded support for the Folding @ home PS3 client and other services available under Life with PlayStation . Over its lifetime of five years and 7 months , more than 15 million users contributed over 100 million hours of computation to Folding @ home , greatly assisting the project with disease research . Following discussions with the Pande lab , Sony decided to terminate the application . Pande considers the PlayStation 3 client a " game changer " for the project .

== = Multi @-@ core processing client == =

Folding @ home can use the parallel computing abilities of modern multi @-@ core processors . The ability to use several CPU cores simultaneously allows completing the full simulation far faster . Working together , these CPU cores complete single work units proportionately faster than the standard uniprocessor client . This approach is scientifically valuable because it enables much longer simulation trajectories to be performed in the same amount of time , and reduces the traditional difficulties of scaling a large simulation to many separate processors . A 2007 publication in the Journal of Molecular Biology relied on multi @-@ core processing to simulate the folding of part of the villin protein approximately 10 times longer than was possible with a single @-@ processor client , in agreement with experimental folding rates .

In November 2006 , first @-@ generation symmetric multiprocessing (SMP) clients were publicly released for open beta testing , referred to as SMP1 . These clients used Message Passing Interface (MPI) communication protocols for parallel processing , as at that time the GROMACS cores were not designed to be used with multiple threads . This was the first time a distributed computing project had used MPI . Although the clients performed well in Unix @-@ based operating systems such as Linux and OS X , they were troublesome under Windows . On January 24 , 2010 , SMP2 , the second generation of the SMP clients and the successor to SMP1 , was released as an open beta and replaced the complex MPI with a more reliable thread @-@ based implementation .

SMP2 supports a trial of a special category of " bigadv " work units , designed for simulating proteins that are unusually large and computationally intensive and have a great scientific priority . These units originally required a minimum of eight CPU cores , which was later increased on February 7 , 2012 to sixteen CPU cores . In addition to these additional hardware requirements over standard SMP2 work units , they require more system resources such as random @-@ access memory (RAM) and Internet bandwidth . In return , users who run these are rewarded with a 20 % increase over SMP2 's bonus point system . The bigadv category allows Folding @ home to run particularly demanding simulations on long timescales that had formerly required the use of supercomputing clusters and could not be performed anywhere else on Folding @ home . Many users with hardware capable of running bigadv units have later had their hardware setup deemed ineligible for bigadv work units when CPU core minimums were increased , leaving them only able to

run the normal SMP work units . This frustrated many users who invested significant amounts of money into the program only to have their hardware be obsolete for bigadv purposes shortly after . As a result , Vijay Pande announced in January 2014 that the bigadv program would end on January 31 , 2015 .

===== V7 =====

The V7 client is the seventh and latest generation of the Folding @ home client software , and is a complete rewrite and unification of the prior clients for Microsoft Windows , OS X and Linux operating systems . It was released on March 22 , 2012 . Like its predecessors , V7 can run Folding @ home in the background at a very low priority , allowing other applications to use CPU resources as they need . It is designed to make the installation , start @-@ up , and operation more user @-@ friendly for novices , as well as offer greater scientific flexibility to researchers than prior clients . V7 uses Trac for managing its bug tickets so that users can see its development process and provide feedback .

V7 consists of four integrated elements . The user typically interacts with V7 's open @-@ source GUI , known as FAHControl . This has Novice , Advanced , and Expert user interface modes , and has the ability to monitor , configure , and control many remote folding clients from a single computer . FAHControl directs FAHClient ? a back @-@ end application that in turn manages each FAHSlot (or " slot ") . Each slot acts as replacement for the formerly distinct Folding @ home v6 uniprocessor , SMP , or GPU computer clients , as it can download , process , and upload work units independently . The FAHViewer function , modeled after the PS3 's viewer , displays a real @-@ time 3 @-@ D rendering , if available , of the protein currently being processed .

===== Google Chrome =====

In 2014 , a client for the Google Chrome and Chromium web browsers was released , allowing users to run Folding @ home in their web browser . The client uses Google 's Native Client (NaCl) feature on Chromium @-@ based web browsers to run the Folding @ Home code at near @-@ native speed in a sandbox on the user 's machine .

===== Android =====

In July 2015 , a client for Android mobile phones was introduced , which is available on Google Play .

== Comparison to other molecular simulators ==

Rosetta @ home is a distributed computing project aimed at protein structure prediction and is one of the most accurate tertiary structure predictors . The conformational states from Rosetta 's software can be used to initialize a Markov state model as starting points for Folding @ home simulations . Conversely , structure prediction algorithms can be improved from thermodynamic and kinetic models and the sampling aspects of protein folding simulations . As Rosetta only tries to predict the final folded state , and not how proteins fold , Rosetta @ home and Folding @ home are complementary and address very different molecular questions .

Anton is a special @-@ purpose supercomputer constructed for molecular dynamics simulations . In October 2011 Anton and Folding @ home were the two most powerful molecular dynamics systems . Anton is unique in its ability to produce single ultra @-@ long computationally expensive molecular trajectories , such as one in 2010 which reached the millisecond range . These long trajectories may be particularly helpful towards certain types of biochemical problems . However , Anton does not use Markov state models for analysis . In 2011 , the Pande lab constructed a MSM from two 100 @-@ μ s Anton simulations and found alternative folding pathways that were not visible through Anton 's traditional analysis . They concluded that there was little difference between MSMs

constructed from a limited number of long trajectories or one assembled from many shorter trajectories . In June 2011 Folding @ home began additional sampling of an Anton simulation in an effort to better determine how its techniques compare to Anton 's methods . However , unlike Folding @ home 's shorter trajectories , which are more amenable to distributed computing and other parallelization techniques , longer trajectories do not require adaptive sampling to sufficiently sample the protein 's phase space . Due to this , it is possible that a combination of Anton 's and Folding @ home 's simulation methods would provide a more thorough sampling of this space .