BioNetGen Language (BNGL) Comprehensive Cheatsheet

This is the definitive guide to the **Native BioNetGen Syntax**, covering all blocks, commands, and arguments supported by the core parser and `BNG2.pl`.

---

1. Model Structure

```
begin model
  # Blocks supported by the native parser (BNG2.pl):
  # parameters, compartments, molecule types, species, seed species,
  # observables, functions, energy patterns,
  # population types, population maps,
  # reaction rules, reactions, groups
end model

begin actions
  # Action commands: generate_network, simulate, writeSBML, etc.
end actions

# Optional: a separate protocol block (native) can be defined outside the model.
# It is executed via the `simulate_protocol()` action (and can also be invoked from
↪  `parameter_scan` with method=>"protocol").
begin protocol
  # action: simulate {method=>"ssa", t_end=>10, n_steps=>100}
end protocol
```

---

2. Molecule Types & Patterns

Defines the structure of molecules.

Syntax: `Name(site1, site2~state1~state2)`

```
begin molecule types
  # Simple
  Ligand(r)

  # States (tilde)
  Protein(s~u~p)

  # Component with same names (allowed but requires careful indexing)
  Complex(a, a)
end molecule types
```

Pattern Syntax

Symbol	Meaning	Example
`~`	State	`A(y~p)`
`!`	Specific Bond	`A(b!1).B(a!1)`
`!+`	Bound (any)	`A(b!+)`
`!?`	Wildcard Bond	`A(b!?)`
`.`	Complex	`A().B()`
`%`	Tag/Label	`A()%1`

---

3. Reaction Rules

Syntax: `Reactants -> Products Rate` **Modifiers**: Append to end of rule.

```
begin reaction rules
  # Standard
  A(b) + B(a) <-> A(b!1).B(a!1) k_on, k_off

  # Tagging for specific molecule tracking
  A(b)%1 + B(a) -> A(b!1)%1.B(a!1) k_on

  # MoveConnected: Move entire complex between compartments
  @EC:L(r!1).@PM:R(l!1) -> @PM:L(r!1).@CP:R(l!1) k_trans MoveConnected

  # DeleteMolecules: Remove reactants
  D(a) -> 0 k_deg DeleteMolecules

  # Contextual Constraints
  # include_reactants(index, pattern)
  # exclude_reactants(index, pattern)
  A(s) -> A(s~p) k_cat exclude_reactants(1, A(s!+))
end reaction rules
```

---

4. Action Commands (Comprehensive)

Most actions take their arguments as a hash map `{key=>value}`. Some native actions use positional arguments instead (notably: `setParameter`, `setConcentration`, `addConcentration`, `setVolume`, `readNFspecies`, `quit`).

`generate_network`

Constructs the species and reaction network from rules.

Argument	Description	Default
`overwrite`	Overwrite existing net file	0
`continue`	Continue network generation if network exists	1
`max_iter`	Max iterations of rule application	100
`max_agg`	Max aggregation size	1e9
`max_stoich`	Max stoichiometry (hash; omitted means no limit)	`{}`
`check_iso`	Check graph isomorphism (canonicalization)	1
`prefix`	Output prefix / basename for `.net` output	model output prefix
`suffix`	Appended to `prefix` as `prefix_suffix`	(unset)
`print_iter`	Print progress during generation	0
`TextSpecies`	Write text species file	1
`TextReaction`	Write text reaction file	0
`verbose`	Detailed logging	0
`write`	Write network output	1

```
generate_network({overwrite=>1, max_iter=>50})
```

`simulate`

Runs the simulation.

Common Arguments

Argument	Description
`method`	`"cvode"`, `"ssa"`, `"pla"`, `"psa"`, `"nf"` (also accepts `"ode"` as an alias for `"cvode"`)
`t_start`	Start time (default 0; if `continue=>1` defaults to current model time)
`t_end`	End time
`n_steps`	Number of output points (alias: `n_output_steps`; cannot specify both)
`sample_times`	Explicit time points (array; must contain 3 or more points)
`argfile`	Read arguments from a file; supports `sample_times` lines like `sample_times [5e-1,1,5.0,1E1]`
`prefix`	Output prefix (default: model output prefix)
`suffix`	Appended to output prefix as `prefix_suffix`
`netfile`	Use a specific `.net` file instead of generating/using `prefix.net`
`verbose`	More detailed output
`continue`	Continue from last state (1 or 0)
`print_end`	Print extra info at end (1 or 0)
`print_n_species_active`	Print number of active species (1 or 0)
`print_net`	Print network (same as `save_progress`; cannot define both)
`save_progress`	Save progress (same as `print_net`; preferred)
`seed`	RNG seed (if omitted, a random seed is generated by BNG)
`print_CDAT`	Print CDAT output (1 or 0; default 1)
`print_functions`	Print function values in output (1 or 0; default 0)
`stop_if`	Stop condition expression (string)
`print_on_stop`	Print when `stop_if` triggers (default 1 if `stop_if` is set)
`max_sim_steps`	Maximum simulation steps (stochastic)
`output_step_interval`	Output every N steps (stochastic)
`update_interval`	Update interval (default 1.0)
`expand`	On-the-fly expansion method: `"lazy"` (default) or `"layered"`

ODE Specific Options (CVODE)

Argument	Description	Default
`atol`	Absolute Tolerance	1e-8
`rtol`	Relative Tolerance	1e-8
`sparse`	Use sparse Jacobian (good for large nets)	0
`steady_state`	Run to steady state	0

PLA Specific Options

Argument	Description	Default
`pla_config`	PLA config string passed to `run_network`	'fEuler
`pla_output`	Extra PLA output control	(unset)

PSA / Scaling Options

Argument	Description	Default
`poplevel`	Scaling target (also forces `method=>"psa"` if provided)	`100`
`check_product_scale`	Product scaling check flag	(unset)

NF (Network Free) Options (NFsim)

Argument	Description	Default
`binary_output`	NFsim `-b`	0
`complex`	NFsim `-cb`	1
`equil`	NFsim `-eq <...>`	(unset)
`get_final_state`	NFsim `-ss prefix.species`	1
`gml`	NFsim `-gml <...>`	(unset)
`nocslf`	NFsim `-nocslf`	0
`notf`	NFsim `-notf`	0
`utl`	NFsim `-utl <...>`	(unset)
`param`	Extra raw NFsim CLI args (string, appended)	(unset)

Related native helper (positional args):

```
# Read an NFsim .species file back into the model state
readNFspecies("output_prefix.species")
```

NF caveats:

- `continue=>1` is not supported by NFsim (returns an error).
- `sample_times` is not supported by this version of NFsim.

```
simulate({method=>"ode", t_end=>100, n_steps=>100, atol=>1e-10, sparse=>1})
simulate({method=>"ssa", t_end=>500, suffix=>"stoch"})
simulate({method=>"nf", t_end=>100, n_steps=>100})
```

Direct native simulator actions also exist (these are what `simulate({method=>...})` dispatches to):

```
simulate_ode({t_end=>100, n_steps=>100})
simulate_ssa({t_end=>100, n_steps=>100})
simulate_pla({t_end=>100, n_steps=>100})
simulate_psa({t_end=>100, n_steps=>100})
simulate_nf({t_end=>100, n_steps=>100})
simulate_protocol({})
```

parameter_scan

Systematically varies a parameter.

Argument	Description
parameter	Name of parameter to scan
par_scan_vals	Explicit array of values to scan (if provided, can omit par_min/par_max/n_scan_pts)
par_min	Minimum value (takes precedence over par_scan_vals if par_max/n_scan_pts also set)
par_max	Maximum value
n_scan_pts	Number of points
log_scale	Logarithmic spacing (1 or 0; default 0)
method	Simulation method for each run: any valid simulate method, plus "protocol" to execute a begin protocol block via simulate_protocol()
t_end	Duration of each sim (required by simulate unless using sample_times)
reset_conc	Reset concentrations between runs (default 1). If true, sets get_final_state=>0 (helps NFsim scans).
prefix	Output prefix (default: model output prefix)
suffix	If set, scan basename becomes prefix_suffix (otherwise prefix_parameter)
parallel	Run in parallel using fork (1 or 0; default 0)
num_cores	Number of forked workers when parallel=>1

```
parameter_scan({parameter=>"k_off", par_min=>0.01, par_max=>100, n_scan_pts=>20,
↪  log_scale=>1, method=>"ode", t_end=>100})
```

bifurcate

Bifurcation analysis (requires AUTO binding). Uses parameter_scan internally (forward + backward scans), so it accepts the same scan/sim arguments.

Required:

- parameter, par_min, par_max, n_scan_pts

Common optional:

- log_scale (default 0)
- method, t_end, reset_conc, prefix, suffix, parallel, num_cores, ... (see parameter_scan)

Note: output is written to per-observable *_bifurcation_<Observable>.scan files.

generate_hybrid_model

Constructs a hybrid particle population (HPP) model and writes it to a new BNGL file.

Argument	Description	Default
prefix	Output prefix (base; _<suffix> is appended)	model output prefix
suffix	Suffix appended to output prefix	hpp
overwrite	Overwrite existing output BNGL	0
verbose	More detailed progress messages	0

Argument	Description	Default
`actions`	Actions to append to generated BNGL (array of strings)	`["writeXML()"]`
`execute`	Execute the `actions` on the hybrid model after writing	0
`safe`	"Safe/exact" hybridization mode	0

Compatibility note: `exact` is accepted as an alias for `safe` (deprecated warning).

```
generate_hybrid_model({suffix=>"hpp", overwrite=>1, actions=>["writeXML()"],
↪  execute=>1})
```

`LinearParameterSensitivity`

Brute-force linear sensitivity analysis by bumping parameters and running simulations.

Status note: the source marks this action as **NOT IMPLEMENTED YET** and it contains limitations (e.g., stochastic mode is noted as unsupported).

Common arguments (as implemented):

- `net_file` (base `.net` prefix; defaults to model output prefix)
- `t_end` (**required**)
- `bump` (percent; default 5)
- `stochast` (0/1; noted as not currently handled)
- `sparse` (default 1), `atol`/`rtol` (default `1e-8`)
- `init_equil` (default 1), `t_equil` (default `1e6`), `re_equil` (default 1)
- `n_steps` (default 50), `suffix` (default "")
- `inp_ppert` (`{pnames=>[...], pvalues=>[...]}`), `inp_cpert` (`{cnames=>[...], cvalues=>[...]}`)

`visualize`

Generates graph visualizations.

Valid `type` values (native):

- `ruleviz_pattern` (alias: `conventional`)
- `ruleviz_operation` (alias: `compact`)
- `regulatory` (default)
- `reaction_network`
- `contactmap`
- `process`
- `rinf`
- `opts`

Argument	Description	Default
`help`	Show visualize help	0
`type`	Visualization type	`regulatory`
`opts`	Visualization options file(s); if a single filename is given it is coerced to a list	(unset)
`background`	Background flag	0
`collapse`	Collapse nodes	0
`each`	Visualize each rule individually	0
`groups`	Grouping flag	0
`textonly`	Text-only output	0

Argument	Description	Default
suffix	Output suffix appended to prefix	""
filter	Filter flag	0
level	Detail level	1
mergepairs	Merge pairs flag	0
embed	Embed flag	0
reset	Reset flag	1
ruleNames	Use rule names	0
doNotUseContextWhenGrouping	Grouping control flag	0
removeReactantContext	Context removal flag	0
makeInhibitionEdges	Inhibition edge flag	0
removeProcessNodes	Process node removal flag	0
compressRuleMotifs	Motif compression flag	0
doNotCollapseEdges	Edge collapse control flag	0

```
visualize({type=>"contactmap"})
```

---

5. Input / Output Commands

These actions can appear in **begin actions** and are implemented natively by **BNG2.pl**.

Model / Network Writing (BNGModel)

writeModel Write the BNGL model (defaults: **format=>"bngl"**, **include_model=>1**, **include_network=>0**).

writeNetwork Write the reaction network (NET format; defaults: **format=>"net"**, **include_model=>0**, **include_network=>1**).

writeNET (deprecated) Deprecated convenience action (writes both model and network; defaults include **evaluate_expressions=>1**, **overwrite=>1**, **TextSpecies=>1**). Prefer **writeModel** / **writeNetwork**.

writeFile General writer used by **writeModel**/**writeNetwork**.

Key options:

- **format** one of **bngl**, **net**, **xml** (native **writeFile()** does **not** currently support **sbml** or **ssc**)
- **include_model**, **include_network**
- **prefix**, **suffix**, **overwrite**
- **pretty_formatting**, **evaluate_expressions**
- **TextSpecies**, **TextReaction**

```
writeModel({overwrite=>1})
writeNetwork({overwrite=>1, pretty_formatting=>0})
writeFile({format=>"xml", include_model=>1, include_network=>1, suffix=>"full"})
```

Export Writers (BNGOutput)

Export to additional formats using dedicated **write*** actions (separate from **writeFile**).

```
writeXML({})
writeSBML({})
writeSBMLMulti({})
writeMDL({})
writeSSC({})
```

```
writeSSCcfg({})
writeMfile({})     # MATLAB ODE file
writeMEXfile({})   # MEX C-code for MATLAB
writeMexfile({})   # alias spelling also exists
writeCPPfile({})
writeCPYfile({})
writeLatex({})     # LaTeX output (case/alias variants exist in source)
```

readFile

Read another BNGL/NET file (and SBML `.xml` via the `sbmlTranslator` helper, if available).

```
readFile({file=>"modules/protein_defs.bngl"})
```

Aliases (thin wrappers around `readFile`):

```
readModel({file=>"my_model.bngl"})
readNetwork({file=>"my_model.net"})
```

saveConcentrations / resetConcentrations

Manages simulation state stack.

```
saveConcentrations()   # Push current state
resetConcentrations()  # Pop/Restore last state
```

setConcentration / setParameter

Runtime modification.

```
setConcentration("A(s)", 50)
setParameter("k_cat", 0.5)
```

addConcentration

Increment a species concentration by a value.

```
addConcentration("A(s)", 10)
```

saveParameters / resetParameters

Manages a parameter-definition stack (optionally by label).

```
saveParameters()          # or saveParameters("my_label")
resetParameters()         # or resetParameters("my_label")
```

setVolume

Set a compartment volume.

```
setVolume("cyto", 1.0)
```

quit

Immediate exit from BioNetGen (no cleanup; useful to stop before running actions).

```
quit()
```

---

6. Mathematical Functions

Can be used in Expressions, Rates, and Functions block.

Function	Description
`_pi`	π (built-in constant)
`_e`	e (built-in constant)
`time`	Current simulation time (0 if undefined)
`exp(x)`, `ln(x)`, `log10(x)`, `log2(x)`	Exponential/Logarithmic
`abs(x)`, `rint(x)`, `sqrt(x)`	Absolute / round-to-nearest-int / root
`sin(x)`, `cos(x)`, `tan(x)`	Trigonometric
`asin(x)`, `acos(x)`, `atan(x)`	Inverse trig
`sinh(x)`, `cosh(x)`, `tanh(x)`	Hyperbolic trig
`asinh(x)`, `acosh(x)`, `atanh(x)`	Inverse hyperbolic trig
`if(cond, a, b)`	Conditional
`min(...)`, `max(...)`, `sum(...)`, `avg(...)`	Variadic aggregates
`mratio(x, y, z)`	Native helper function
`TFUN(obs, file)`	NFsim helper; attempting to evaluate outside NFsim errors

---

7. Header & Configuration Blocks

Optional blocks at the beginning of the file.

```
begin model
  # ...
end model

# Set BNGL version
version("2.2.6")
# Version strings support an optional relation and codename suffix:
#   version("MAJOR.MINOR.DIST+")   # (default) require this version or later
#   version("MAJOR.MINOR.DIST-")   # require this version or earlier
#   version("MAJOR.MINOR.DIST+ CODENAME")
# Or require a codename explicitly:
#   codename("CODENAME")

# Override the model name used for output basenames
setModelName("MyModel")

# Set substance units (native values are "Number" or "Concentration")
substanceUnits("Concentration")

# Global options
setOption("SpeciesLimit", 1000)
```

---

8. Advanced Rule Features

Rule Modifiers

Modifier	Description
`DeleteMolecules`	Deletes matched reactants
`MoveConnected`	Moves connected complexes
`TotalRate`	Sets the total rate of the rule explicitly
`include_reactants(i, P)`	Reactant i must match pattern P
`exclude_reactants(i, P)`	Reactant i must NOT match P
`include_products(i, P)`	Product i must match pattern P
`exclude_products(i, P)`	Product i must NOT match P

`MatchOnce` is **not** a rule modifier in native BioNetGen; it is a **pattern attribute** written in curly braces (e.g., `{MatchOnce=1}`) and is not allowed on concrete species.

Tagging & Maps

Use `%x` to tag molecules and track them from reactant to product (ensure isomorphism).

```
A(s)%1 + B(s) -> A(s!1)%1.B(s!1) k_on
```

---

9. Math & Kinetics Functions

In addition to standard expression functions (see section 6), native BioNetGen supports special rate-law forms in reaction rules.

Rate law	Notes
`Sat(p1,p2,...)`	Special RateLaw type `Sat`. Arguments must be identifiers (parameter/function names), not arbitrary expressions.
`MM(p1,p2,...)`	Special RateLaw type `MM`. Arguments must be identifiers.
`Hill(p1,p2,...)`	Special RateLaw type `Hill`. Arguments must be identifiers.
`Arrhenius(phi_expr, actE_expr)`	Two expressions are parsed. Internally stored as two generated local functions (names like `_Aphi_...`, `_AEact0_...`).
`FunctionProduct("expr1","expr2")`	Product of two quoted expressions; each is parsed as an Expression and stored as local functions.

---

10. Advanced Action Options

`simulate` **Extras**

Arg	Description
`stop_if`	Condition to stop simulation: `stop_if=>"A(s~p)>100"`
`print_functions`	Print function values in output (1 or 0)
`sample_times`	List of specific times to record (must contain 3+ points): `[1,10,100]`

`generate_network` **Extras**

Arg	Description
`max_stoich`	Limit max stoichiometry per species: `{A=>10}`
`print_iter`	Print network stats every N iterations

---

11. Resources & Tutorials

- **Official Documentation**
- **BioNetGen Tutorial & Quick Reference**
- **Biological Modeling (Chemotaxis Example)**
- **BNGL Grammar (EBNF)**
- **Video Tutorials:**
  - Introduction to BioNetGen
  - Advanced Usage
- **Workshop Slides (2021):**
  - Intro to Rxn Net Modeling
  - Intro to Rule-Based Modeling
- **Key Publications:**
  - BioNetGen 2.2: Advances in Rule-Based Modeling (Bioinformatics)
  - Rule-Based Modeling Protocol (Springer)

---

12. Published BioNetGen Models

A selection of publications applying BioNetGen to biological systems.

2023

- **Korwek Z et al.** Non-self RNA rewires IFNβ signaling: A mathematical model of the innate immune response. *Sci. Signaling*
- **Zhang Y et al.** Combining Multikinase Tyrosine Kinase Inhibitors... *ACS Pharmacol. Transl. Sci.*

2022

- **Nosbisch JL et al.** A kinetic model of phospholipase C-γ1... *J. Biol. Chem.*

2021

- **McMillan D et al.** Structural insights into the disruption of TNF-TNFR1 signalling... *Nat. Commun.*
- **Zhang Y et al.** A systems biology model of junctional localization and downstream signaling of the Ang-Tie signaling pathway. *NPJ Syst. Biol. Appl.*
- **Erdem C et al.** Inhibition of RPS6K reveals context-dependent Akt activity... *PLOS Comput. Biol.*

2020

- **Bolado-Carrancio A et al.** Periodic propagating waves coordinate RhoGTPase network dynamics... *Elife*
- **Ordyan M et al.** Interactions between calmodulin and neurogranin govern the dynamics of CaMKII... *PLOS Comput. Biol.*
- **Kirsch K et al.** Co-regulation of the transcription controlling ATF2 phosphoswitch by JNK and p38. *Nat. Commun.*
- **Salzer B et al.** Engineering AvidCARs for combinatorial antigen recognition... *Nat. Commun.*

- **Wu Q, Finley SD.** Mathematical Model Predicts Effective Strategies to Inhibit VEGF-eNOS Signaling. *J. Clin. Med.*
- **Kapralov AA et al.** Redox lipid reprogramming commands susceptibility of macrophages... *Nat. Chem. Biol.*

2019

- **Erickson KE et al.** Modeling cell line-specific recruitment of signaling proteins to the insulin-like growth factor 1 receptor. *PLOS Comput. Biol.*
- **Nikolaev Y et al.** Systems NMR: single-sample quantification of RNA, proteins and metabolites... *Nat. Methods*
- **Lin YT, Feng S, Hlavacek WS.** Scaling methods for accelerating kinetic Monte Carlo simulations... *J. Chem. Phys.*

2018

- **Wong VC et al.** NF-κB-Chromatin Interactions Drive Diverse Phenotypes... *Cell Reports*
- **Bazzazi J et al.** Computational modeling of synergistic interaction between αVβ3 integrin and VEGFR2... *J. Theor. Biol.*
- **Tse MJ et al.** Rare-event sampling of epigenetic landscapes and phenotype transitions. *PLOS Comput. Biol.*
- **Singh M et al.** Shift from stochastic to spatially-ordered expression of serine-glycine synthesis enzymes... *Sci. Rep.*
- **Rohrs JA et al.** Computational Model of Chimeric Antigen Receptors Explains Site-Specific Phosphorylation Kinetics. *Biophys. J.*
- **James JR.** Tuning ITAM multiplicity on T cell receptors can control potency and selectivity... *Sci. Signal*
- **Czerkies M et al.** Cell fate in antiviral response arises in the crosstalk of IRF, NF-κB and JAK/STAT pathways. *Nat. Commun.*
- **Rukhlenko OS et al.** Dissecting RAF Inhibitor Resistance by Structure-based Modeling... *Cell Syst.*

2017

- **Harmon B et al.** Timescale Separation of Positive and Negative Signaling... *Sci. Rep.*
- **Meng X et al.** Minimum-noise production of translation factor eIF4G... *Nucleic Acids Res.*
- **Bazzazi J et al.** Inhibition of VEGFR2 Activation and Its Downstream Signaling... *Front Physiol.*

2016

- **Antunes G et al.** Modelling intracellular competition for calcium... *Sci. Rep.*
- **Hat B et al.** Feedbacks, Bifurcations, and Cell Fate Decision-Making in the p53 System. *PLOS Comput. Biol.*
- **Rohrs JA et al.** Predictive model of thrombospondin-1 and vascular endothelial growth factor... *NPJ Syst. Biol. Appl.*
- **Korwek Z et al.** Importins promote high-frequency NF-κB oscillations... *Biol. Direct*
- **Camillo BD et al.** A rule-based model of insulin signalling pathway. *BMC Syst. Biol.*

2015

- **Dolan DWP et al.** Integrated Stochastic Model of DNA Damage Repair by Non-homologous End Joining... *PLOS Comput. Biol.*
- **Stites EC et al.** Use of mechanistic models to integrate and analyze multiple proteomic datasets. *Biophys. J.*
- **Hawse WF et al.** Cutting Edge: Differential Regulation of PTEN by TCR, Akt, and FoxO1... *J. Immunol.*
- **Szymańska P et al.** Computational analysis of an autophagy/translation switch... *PLOS One*

2014

- **Ligon TS et al.** Multi-level kinetic model of mRNA delivery via transfection of lipoplexes. *PLOS One*
- **Chylek LA et al.** Phosphorylation site dynamics of early T-cell receptor signaling. *PLOS One*
- **Dushek O et al.** Biosensor architectures for high-fidelity reporting of cellular signaling. *Biophys. J.*

2013

- **Ibrahim B et al.** Spatial rule-based modeling: a method and its application to the human mitotic kinetochore. *Cells*
- **Pękalski J et al.** Spontaneous NF-κB activation by autocrine TNFα signaling... *PLOS One*
- **Barua D, Hlavacek WS.** Modeling the effect of APC truncation on destruction complex function... *PLOS Comput. Biol.*
- **Fengos G, Iber D.** Prediction stability in a data-based, mechanistic model of σF regulation... *Sci. Rep.*
- **Mukhopadhyay H et al.** Systems model of T cell receptor proximal signaling reveals emergent ultrasensitivity. *PLOS Comput. Biol.*
- **Falkenberg CV, Loew LM.** Computational analysis of Rho GTPase cycling. *PLOS Comput. Biol.*

2012

- **Barua D, Goldstein B.** A mechanistic model of early FcεRI signaling... *PLOS One*
- **Creamer MS et al.** Specification, annotation, visualization and simulation of a large rule-based model for ERBB receptor signaling. *BMC Syst. Biol.*
- **Barua D et al.** A computational model for early events in B cell antigen receptor signaling... *J. Immunol.*
- **Michalski PJ, Loew LM.** CaMKII activation and dynamics are independent of the holoenzyme structure... *Phys. Biol.*

2011

- **Geier F et al.** A computational analysis of the dynamic roles of talin, Dok1, and PIPKI for integrin activation. *PLOS One*
- **Dushek O et al.** Ultrasensitivity in multisite phosphorylation of membrane-anchored proteins. *Biophys. J.*
- **Thomson TM et al.** Scaffold number in yeast signaling system sets tradeoff between system output and dynamic range. *PNAS*

2010

- **Nag A et al.** Shaping the response: the role of FcεRI and Syk expression levels... *IET Syst. Biol.*
- **Monine MI et al.** Modeling multivalent ligand-receptor interactions with steric constraints... *Biophys. J.*
- **Ray JC, Igoshin OA.** Adaptable functionality of transcriptional feedback in bacterial two-component systems. *PLOS Comput. Biol.*

2009

- **Dushek O et al.** A role for rebinding in rapid and reliable T cell responses to antigen. *PLOS Comput. Biol.*
- **Barua D et al.** A bipolar clamp mechanism for activation of Jak-family protein tyrosine kinases. *PLOS Comput. Biol.*
- **An GC, Faeder JR.** Detailed qualitative dynamic knowledge representation using a BioNetGen model... *Math. Biosci.*
- **Nag A et al.** Aggregation of membrane proteins by cytosolic cross-linkers... *Biophys. J.*

2008 & Earlier

- **Barua D et al. (2008)** Computational models of tandem SRC homology 2 domain interactions... *J. Biol. Chem.*
- **Mu F et al. (2007)** Carbon-fate maps for metabolic reactions. *Bioinformatics*
- **Barua D et al. (2007)** Structure-based kinetic models of modular signaling protein function: focus on Shp2. *Biophys. J.*
- **Blinov ML et al. (2006)** A network model of early events in epidermal growth factor receptor signaling... *Biosystems*
- **Faeder JR et al. (2005)** Combinatorial complexity and dynamical restriction of network flows... *IET Syst. Biol.*
- **Faeder JR et al. (2003)** Investigation of early events in FcεRI-mediated signaling... *J. Immunol.*